สำนักหอสมุดกลาง
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การค้นหาภาพแผนที่แสดงเส้นทางจราจรกรุงเทพมหานครโดยใช้ฐานความรู้

A Bangkok Traffic Map Retrieval by Rule-Based System

โดย

นายวรฉัตร ธรรมอารี

วัน เดือน ปี 14 ม.ค. 2539
เลขทะเบียน 034938
เลขเรียกหนังสือ T371.334 W4

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต

ภาควิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034938

ชื่อเรื่องปริญญานิพนธ์ การค้นหาภาพแผนที่แสดงเส้นทางจราจรกรุงเทพมหานครโดยใช้ฐานความรู้

A Bangkok Traffic Map Retrieval by Rule-Based System

โดย

นายวรฉัตร ธรรมอารี 34106302

อาจารย์ที่ปรึกษา

รศ.ครรชิต ไมตรี

ปริญญานิพนธ์นี้สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การค้นหาภาพแผนที่แสดงเส้นทางการจราจรกรุงเทพมหานครโดยใช้ฐานความรู้
(A Bangkok Traffic Map Retrieval by Rule-Based System)

ผู้จัดทำ นายวรฉัตร ธรรมอารี

..... อาจารย์ที่ปรึกษา
(อ. ครรชิต ไมตรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาภาพแผนที่แสดงเส้นทางจราจรกรุงเทพมหานครโดยใช้ฐานความรู้

วรฉัตร ธรรมอารี

รศ.ดร.ครรชิต ไมตรี อาจารย์ที่ปรึกษา

ปีการศึกษา 2537

บทคัดย่อ

ปัจจุบันสิ่งที่เป็นปัญหาของคนส่วนมากที่อาศัยอยู่ในกรุงเทพมหานครก็คือปัญหาการจราจะที่ติดขัด ซึ่งการแก้ปัญหานี้มีอยู่หลายวิธี แต่มีอยู่วิธีหนึ่งที่ช่วยในการเลี่ยงปัญหาการจราจรที่ติดขัดก็คือ การศึกษาหาเส้นทางลัดต่างๆ เพื่อใช้ในการเดินทาง ซึ่งแท้จริงแล้วการเดินทางในเส้นทางลัดอาจไม่ใช่วิธีการเดินทางที่ดีที่สุดก็ได้ เพราะอาจจะใช้เวลาในการเดินทางมากขึ้นก็ได้ ดังนั้นนอกจากจะดูที่เส้นทางลัดแล้ว เรายังต้องคำนึงถึงปริมาณรถในแต่ละเส้นทางและเวลาที่ใช้ในการเดินทางในแต่ละเส้นทางลัดนั้นๆ การศึกษาค้นหาเส้นทางในการเดินทางที่ดีที่สุดจึงเกิดขึ้นโดยใช้แนวคิดที่ว่า ใช้เวลาในการเดินทางให้สั้นที่สุด แต่ทว่าการศึกษาในหัวข้อนี้มีปัจจัยที่ไม่แน่นอนเกี่ยวข้องมากมาย เช่น ถ้าถนนเส้นนั้นเกิดอุบัติเหตุ, ปิดถนนเพื่อซ่อมแซมถนน เป็นต้น ดังนั้นวิทยานิพนธ์นี้จึงเป็นเพียงแค่การทดลองศึกษาโดยการรวบรวมวิธีการต่างๆเข้ากับปัจจัยหลักๆต่างๆที่เกี่ยวข้องมาประยุกต์เข้าด้วยกัน พร้อมกันเสนอวิธีการปรับปรุงและแนวทางในการพัฒนาโปรแกรมให้มีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Bangkok Traffic Map Retrieval by Rule-Based System

Vorachat Tamarree

Associate Professor Kanchit Maitree Advisor

1994

Abstract

In this day, the most serious problem of people that live in Bangkok is the traffic jam problem. There are many ways to solve this problem. The most popular way to solve this problem is to find any other shorter path (road, troake, soi), although it may use more time to go in that path. Not only to look up for the shortest path, but we also have to think about the car's volume in each path and the time that use to pass that path. So, the study to find the best path was occured by the way that use the least time. To study in this topic has many uncertain factors such as have the car-accident in any path, close the road to repair it and so on. This thesis is the experimentation by using many solving way to find the path applying with many major factor. And, this thesis, give the efficient way to improve and develop program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อวิทยานิพนธ์ภาษาไทย		ก
บทคัดย่อวิทยานิพนธ์ภาษาอังกฤษ		ข
สารบัญรูป		ค
สารบัญตาราง		ง
บทที่		
1.	บทนำ	1
	1.1 ที่มาและความสำคัญ	1
	1.2 วัตถุประสงค์ของการศึกษา	2
	1.3 ขอบเขตการศึกษา	2
	1.4 แหล่งที่มาของข้อมูล	3
	1.5 วิธีการดำเนินงาน	4
	1.6 ประโยชน์ที่คาดว่าจะได้รับ	5
	1.7 ปัญหาต่างๆและแนวทางการแก้ปัญหา	5
2.	ทฤษฎีและหลักการที่ใช้	7
	2.1 เทคนิคในการค้นหา	7
	2.1.1 การประเมินค่าการค้นหาแต่ละวิธี (Evaluating a Search)	8
	2.1.2 (Uninformed State Space Search (แบบไม่มี Heuristic))	9
	2.1.2.1 Depth-First Search	9
	อัลกอริทึม	10
	การประเมินค่า	11
	ข้อดี	11
	2.1.2.2 Breadth-First Search	12
	อัลกอริทึม	12
	การประเมินค่า	13
	ข้อดี	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

2.1.3 (Informed State Space Search (แบบ Heuristic)) 14

2.1.3.1 ลักษณะปัญหาที่ควรใช้ Heuristic 15

2.1.3.2 Hill-Climbing 16

อัลกอริทึม 17

การประเมิน 17

2.1.3.3 Least-Cost 18

การประเมิน 18

2.1.4 การเลือกว่าควรจะใช้วิธีในการค้นหาแบบไหน 19

2.2 ส่วนการวาดภาพแผนที่โดยใช้ Digitizer 19

2.2.1 ทำไมต้อง Digitizer 20

2.3 ส่วนการเขียนโปรแกรม AutoLISP 21

2.3.1 อะไรคือ AutoLISP 21

2.3.2 โครงสร้างของ AutoLISP 21

2.3.3 องค์ประกอบของ AutoLISP 23

2.3.4 การเขียนและการใช้งาน AutoLISP 27

2.3.5 การใช้งาน AutoLISP บน AutoCAD 29

2.4 การนำโปรแกรมค้นหาเส้นทางมาประยุกต์เข้ากับภาพแผนที่ 34

2.4.1 อธิบายแนวคิด 35

3. วิธีการดำเนินงาน 37

3.1 การรวบรวมข้อมูล 38

3.2 การติดตั้ง AutoLISP 40

3.2.1 การติดตั้งพื้นที่หน่วยความจำใน AutoLISP 40

3.2.2 การตั้งค่า BUFFER และ FILE ใน DISK 41

3.2.3 การติดตั้ง Editor ให้ AutoCAD 41

3.2.4 การเขียนโปรแกรมเชื่อมต่อการทำงาน 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

3.3 การเขียนโปรแกรมภาษา Prolog 45

3.3.1 การเขียนโปรแกรมการค้นหาแบบตามแนวลึกก่อน (Depth-First Search) 45

3.3.2 การเขียนโปรแกรมการค้นหาแบบตามแนวกว้างก่อน (Breadth-First Search) 55

3.3.3 การเขียนโปรแกรมการค้นหาแบบปีนตามเนินเขา (Hill Climbing Search) 58

3.3.4 การเขียนโปรแกรมการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด (Least Cost Search) 62

3.3.5 สรุปผลการทำงานรวมของการค้นหาแบบวิธีต่างๆ ทั้ง 4 วิธีที่ผ่านมา 66

3.4 การ digitize ภาพแผนที่ 70

3.4.1 การติดตั้งเครื่อง digitizer 70

3.4.2 การใช้งานเครื่อง digitizer 71

3.5 การรวมเอาส่วนโปรแกรมและเอาส่วนของภาพแผนที่เข้ามาประกอบกัน 72

4. ผลการวิเคราะห์ 74

4.1 ผลที่ได้จากการใช้การค้นหาแบบตามแนวลึกก่อน 76

4.2 ผลที่ได้จากการใช้การค้นหาแบบตามแนวกว้างก่อน 77

4.3 ผลที่ได้จากการใช้การค้นหาแบบปีนตามเนินเขา 78

4.4 ผลที่ได้จากการใช้การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด 79

4.5 สรุปผลของแต่ละวิธี 80

5. สรุปผลการวิเคราะห์และข้อเสนอแนะ 81

5.1 สรุปผลการวิเคราะห์ 81

5.2 ข้อเสนอแนะ 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

ภาคผนวก

ก. คำสั่งต่างๆบางคำสั่งที่จำเป็นในการ Digitize 84

ข. รายชื่อโหนดและค่าโคออร์ดิเนตของแต่ละโหนด 108

ค. ขนาดแผนที่ทั้งหมด และขอบเขตแผนที่ที่ใช้แสดง 123

ง. ภาพแสดงผลลัพธ์ที่เป็นผลจากการค้นหาเส้นทางในแต่ละวิธีเมื่อนำมาใช้กับแผนที่ทั้งหมด 125

กิตติกรรมประกาศ 129

หนังสืออ้างอิง 130

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	แสดงประเภทของการค้นหาทั้ง 2 ประเภท	9
2.2	แสดงตัวอย่างทางเดินในการค้นหาแบบตามแนวลึกก่อน	9
2.3	แสดงภาพการค้นหาแบบตามแนวกว้างก่อน	12
2.4	แสดงถนนที่ใช้เป็นตัวอย่างแนวคิดของวิทยานิพนธ์	35
2.5	แสดงถึงการแปลงจากภาพจราจรเป็นแผนภาพข่ายงาน	35
3.1	แสดงตัวอย่างเส้นทางการเดินทางแบบที่ 1	45
3.2	แสดงตัวอย่างเส้นทางการเดินทางแบบที่ 2	66
4.1	แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบตามแนวลึกก่อน	76
4.2	แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบตามแนวกว้างก่อน	77
4.3	แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบปีนตามเนินเขา	78
4.4	แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด	79
ค.1	แสดงแผนที่ที่ตัดออกมาเพื่อใช้กับโปรแกรมค้นหาเส้นทาง	123
ค.2	แสดงแผนที่รวมทั้งหมดของถนนเยาวราช	124
ง.1	แสดงตัวอย่างการค้นหาแบบตามแนวลึกก่อนกับแผนที่ทั้งหมด	125
ง.2	แสดงตัวอย่างการค้นหาแบบตามแนวกว้างก่อนกับแผนที่ทั้งหมด	126
ง.3	แสดงตัวอย่างการค้นหาแบบปีนตามเนินเขากับแผนที่ทั้งหมด	127
ง.4	แสดงตัวอย่างการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุดกับแผนที่ทั้งหมด	128

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	แสดงเครื่องหมายดำเนินการและความหมาย	26
3.1	แสดงผลการเปรียบเทียบการค้นหาเส้นทางโดยใช้วิธีการต่างๆกับตัวอย่างแบบที่ 1	66
3.2	แสดงผลการเปรียบเทียบการค้นหาเส้นทางโดยใช้วิธีการต่างๆกับตัวอย่างแบบที่ 2	68
4.1	แสดงผลการสรุปวิธีการค้นหาเส้นทางที่ใช้วิธีการต่างๆ	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันสิ่งที่เป็นปัญหามากที่สุดอย่างหนึ่งของคนกรุงเทพมหานคร คือ การจราจรที่ติดขัดด้วยปริมาณรถที่คับคั่งมากขึ้นทุกวันแต่ปริมาณถนนนั้นเท่าเดิม ทำให้เกิดความเสียหายทางเศรษฐกิจทั้งในส่วนบุคคลและประเทศโดยรวม ตลอดจนสร้างความเครียดทางจิตใจ ทำให้เกิดความเสื่อมโทรมทางด้านร่างกายและจิตใจ ซึ่งรัฐบาลได้พยายามแก้ปัญหานี้เป็นเวลานานแล้ว แต่เนื่องจากปริมาณรถที่มากขึ้นทุกวัน ทำให้การแก้ปัญหาไม่ประสบผลสำเร็จเท่าที่ควร

สิ่งที่คนกรุงเทพมหานครพยายามทำเพื่อหลีกเลี่ยงปัญหาจราจรคือ การพยายามหาวิธีการต่างๆเพื่อใช้เวลาในการเดินทางให้น้อยที่สุดจึงเป็นส่วนที่สำคัญ ดังนั้นวิธีการแก้ปัญหาต่างๆจึงเกิดขึ้น เช่น การเลี่ยงช่วงเวลาที่จราจรคับคั่งโดยออกเดินทางก่อนแต่เนิ่นๆ การเลี่ยงการใช้รถยนต์ไปใช้รถมอเตอร์ไซด์แทน เป็นต้น แต่วิธีหนึ่งที่เป็นที่นิยมมากที่สุดวิธีหนึ่งคือ การศึกษาเพื่อหาเส้นทางลัดต่างๆ เพื่อช่วยลดระยะทางในการเดินทาง ซึ่งอันที่จริงแล้วการเดินทางในเส้นทางลัดนั้นอาจจะไม่ใช่วิธีการเดินทางที่ดีที่สุดจริงๆ เพราะสิ่งที่ทุกคนต้องการไม่ใช่ระยะทางที่สั้นที่สุดแต่เป็นเวลาที่ใช้ในการเดินทางที่สั้นที่สุด ดังนั้นการเดินทางในเส้นทางลัดก็อาจจะไม่ทำให้ลดระยะเวลาในการเดินทางเลยก็ได้

ปัจจัยที่ต้องคำนึงถึงนอกจากปัจจัยทางด้านปริมาณรถบนท้องถนนแล้ว ปัจจัยทางข้อมูลเส้นทางลัด สภาพท้องถนน ช่วงเทศกาล และสิ่งต่างๆอีกมากมาย ต่างต้องนำมาประกอบกันเพื่อเป็นข้อมูลในการตัดสินใจว่าควรจะเดินทางในเส้นทางใดเพื่อจะได้ใช้เวลาในการเดินทางให้น้อยที่สุด ซึ่งในปัจจุบันเรามีเครื่องคอมพิวเตอร์เพื่อช่วยในการวิเคราะห์ต่างๆและช่วยในการตัดสินใจ แต่นอกจากเครื่องคอมพิวเตอร์แล้ว การหาวิธีการที่จะให้เครื่องคอมพิวเตอร์คิดเพื่อที่จะตัดสินใจได้ก็มีความสำคัญอยู่มาก เพราะถ้าการหาวิธีการในการค้นหาเส้นทางไม่ดีแล้ว ก็อาจจะไม่ได้ช่วยลดเวลาในการเดินทางเลย ดังนั้นการศึกษาค้นคว้าต่างๆเกี่ยวกับข้อมูลที่เชื่อมต่อกันเป็นโครงข่าย(Network) หรือเป็นกราฟ(Graph) ก็ถูกนำมาประยุกต์ใช้กับแผนที่กรุงเทพมหานคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาในที่นี้จึงเน้นหนักไปที่การศึกษาวิธีการแก้ปัญหาการเดินทางให้ใช้เวลาในการเดินทางให้น้อยที่สุดโดยไม่จำเป็นว่าจะต้องเป็นเส้นทางลัดหรือไม่ก็ได้ โดยใช้ความรู้ความสามารถทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) แบบหนึ่งก็คือ ฐานความรู้ เพื่อนำมาประยุกต์ในการค้นหาเส้นทางต่างๆ และใช้โปรแกรมทางด้าน CAD/CAM ที่ชื่อว่า AutoCad เพื่อให้สามารถเป็นภาพแผนที่ในการเดินทางจริงๆได้ แต่เนื่องจากข้อจำกัดทางด้านเวลาเพราะต้องใช้เวลาในการเก็บแผนที่มาก ดังนั้นในที่นี้จะเป็นการเก็บตัวอย่างแผนที่ถนนบริเวณถนนเยาวราช ที่เลือกบริเวณนี้เนื่องจากเป็นบริเวณชุมชนที่มีถนน ตรอก ซอย มากมายและเป็นตัวอย่างที่ดีในการประยุกต์เข้ากับแผนที่ส่วนอื่นของกรุงเทพมหานครได้เป็นอย่างดี

1.2 วัตถุประสงค์

1. เพื่อศึกษาถึงความเป็นไปได้ที่จะใช้เส้นทางลัด เช่น ตรอก, ซอย เป็นต้น
2. เพื่อที่จะประหยัดเวลาในการเดินทาง และเพื่อการกำหนดเส้นทางเดินทาง โดยใช้เส้นทางต่างๆมาประยุกต์เข้ากับจำนวนรถในแต่ละเส้นทาง
3. เพื่อประยุกต์ใช้ฐานความรู้ให้เกิดประโยชน์กับเส้นทางการจราจรที่มีอยู่ให้มีประสิทธิภาพมากที่สุด

1.3 ขอบเขตการศึกษา

การศึกษานี้เป็นการศึกษาค้นหาเส้นทางในการเดินทางระหว่างโหนดสองโหนด คือโหนดเริ่มต้นและโหนดปลายทาง โดยพิจารณาจากเวลาที่ใช้ในการเดินทางเป็นหลัก ซึ่งได้กำหนดขอบเขตของการศึกษาไว้พอสังเขปดังนี้

1. การเดินทางที่กล่าวถึงนี้ เป็นการพิจารณาเฉพาะการเดินทางโดยใช้รถมอเตอร์ไซด์เท่านั้น ซึ่งหมายรวมถึงเส้นทางทุกเส้นทางไม่ว่าเป็น ถนน, ตรอก, ซอย (ยกเว้น ซอยตัน และ ตรอกตัน) โดยมีการแสดงข้อมูลว่าบริเวณแยกถนนไหน ชื่ออะไร เพื่อเป็นประโยชน์ต่อการเดินทางของผู้เดินเท้าด้วย
2. เส้นทางที่ใช้ในการพิจารณา จะพิจารณาจากเส้นทางจริงที่แสดงไว้ในแผนที่กรุงเทพมหานคร ของกองบังคับการตำรวจจราจร โดยพิจารณาการเดินทางเป็นการวิ่งจากแยกถนนหนึ่ง ไปอีกแยกถนนหนึ่ง เช่น การวิ่งจากสี่แยกหนึ่งไปอีกสี่แยกหนึ่ง หรือการวิ่งจากสามแยกหนึ่งไปอีกสามแยกหนึ่ง เป็นต้น
3. การอ้างอิงจะอ้างอิงได้จากโหนดที่มีชื่อปรากฏในแผนที่เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ทิศทางการเดินรถที่ใช้ ได้พิจารณาถึงทิศทางจริงในการเดินรถบนถนนแต่ละเส้น คือ ทั้งการเดินรถทางเดียว (One Way) และการเดินรถสวนทาง (Two Way) โดยรถมอเตอร์ไซด์จะต้องวิ่งตามเส้นทางเดินรถ นั่นคือห้ามวิ่งสวนทาง และวิ่งบนทางเท้า

5. ทิศทางเดินรถที่กล่าวถึงในข้อ 4 ได้กำหนดให้มีผลบังคับใช้ตลอดทั้งวัน คือ มิได้กำหนดเฉพาะบางช่วงเวลาที่มีผลบังคับใช้

6. อัตราเร็วรถที่ใช้จะกำหนดให้เป็น 30 กิโลเมตร/ชั่วโมง โดยได้ค่านี้มาจากอัตราเร็วเฉลี่ยของรถมอเตอร์ไซด์ที่วิ่งในถนนเยาวราช ซึ่งเป็นอัตราการประมาณที่หาจากระยะทางจริงหารด้วยเวลาที่ใช้ในการวิ่ง ค่าที่ได้จะนำมาหารเฉลี่ย 10 ครั้ง แล้วปัดเศษทิ้ง ซึ่งผลลัพธ์ที่ได้จะได้ประมาณ 30 กิโลเมตร/ชั่วโมง

7. การศึกษานี้พิจารณาถึงเวลาที่รถวิ่งจริงๆในช่วงเวลาการจราจรติดขัด โดยอยู่ในช่วงเวลา 8.00 - 20.00 น. กล่าวคือเป็นการประมาณเท่านั้นว่าควรจะวิ่งไปเส้นไหน ในช่วงเวลาที่การจราจรติดขัด

8. การศึกษานี้ไม่ได้คำนึงถึงปัจจัยในเรื่องการเดินทางวันธรรมดาและวันหยุด เนื่องจากไม่มีหน่วยงานใดเก็บข้อมูล แต่ในความเป็นจริงมีสภาพการจราจรแตกต่างกัน

9. ค่าที่จะนำมาพิจารณาว่าช่วงเส้นทางระหว่างจุดแยกของถนน(ตรอก, ซอย) ไหน รถควรจะวิ่งหรือไม่ (ถ้าค่ามากแสดงว่าการจราจรในถนนดังกล่าวมาก ถ้าค่าดังกล่าวมีค่าน้อยก็แสดงว่าไม่ค่อยมีการจราจรติดขัด ดังนั้นรถควรจะวิ่งในเส้นทางนี้) นำมาจากการหาค่าประมาณของสูตร

$$T = (S/V) * W$$

เมื่อ T คือเวลาที่ใช้ในการวิ่งซึ่งก็คือค่าที่จะนำมาพิจารณา

S คือระยะทางระหว่างโหนด 2 โหนดใดๆ

V คือความเร็วของรถมอเตอร์ไซด์ (มีค่า 30 กิโลเมตรต่อชั่วโมง)

W มีค่าเป็น
- 1 เมื่อเป็นเส้นทางที่มีสภาพการจราจรคล่องตัว
- 1.5 เมื่อเป็นเส้นทางที่มีสภาพการจราจรติดขัดบ้างบางครั้ง
- 2 เมื่อเป็นเส้นทางที่มีสภาพการจราจรติดขัดเกือบตลอดเวลา

10. โหนด คือ บริเวณแยกถนนต่างๆ เช่น สี่แยก สามแยก เป็นต้น และจุดถนนที่รถเลี้ยวได้

1.4 แหล่งที่มาของข้อมูล

1. แผนที่ที่ใช้ในการวิเคราะห์ ใช้แผนที่เส้นทางเดินรถในเขตกรุงเทพมหานคร ซึ่งได้มีการสำรวจ จัดทำ และจัดจำหน่าย โดย บริษัท บางกอกไกด์ จำกัด

2. ข้อมูลของถนน และเส้นทางต่างๆในการเดินทาง รวมทั้ง ตำแหน่งของสถานที่ต่างๆบนเส้นทางเหล่านั้น ได้จากตำแหน่งในแผนที่เส้นทางเดินรถในเขตกรุงเทพมหานคร ประกอบกับแผนที่บางส่วนของกองรังวัดและจัดกรรมสิทธิ์ที่ดิน สำนักการโยธา กรุงเทพมหานคร และจากการสอบถามไปยังสถานที่ดังกล่าวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทิศทางเดินรถของเส้นทางต่างๆที่ได้จัดให้มีการเดินรถทางเดียว (One Way) และเส้นทางที่มีการเดินรถสวนทาง (Two Way) ซึ่งมีผลบังคับใช้ตลอดวัน ได้จากข้อมูลที่รวบรวมโดยกองบังคับการตำรวจจราจร

4. ระยะทางระหว่างจุดแยกของถนนต่างๆ ได้จากการวัดระยะทางในแผนที่เทียบกับอัตราส่วนจริงจากแผนที่ ของ กองรังวัดและจัดกรรมสิทธิ์ที่ดิน สำนักการโยธา กรุงเทพมหานคร ,แผนที่เส้นทางเดินรถในเขตกรุงเทพมหานคร และจากการออกไปสำรวจเก็บข้อมูลเอง

5. ข้อมูลเกี่ยวกับความกว้างของถนนและจำนวนเลนของถนน ได้จากการออกไปรวบรวมข้อมูลจากสถานที่จริงๆเอง

6. ข้อมูลเกี่ยวกับระยะเวลาที่ใช้ในการเดินทาง ได้จากการคำนวณดังที่ได้กล่าวมาแล้ว

7. ข้อมูลเกี่ยวกับเส้นทางลัด ได้จากเส้นทางลัดที่ระบุในแผนที่เส้นทางเดินรถในเขตกรุงเทพมหานครและข้อมูลที่รวบรวมโดย กองการจราจร สำนักการจราจรและขนส่งกรุงเทพมหานคร รวมทั้งในส่วนที่ระบุในเอกสารประชาสัมพันธ์เส้นทางลัดในกรุงเทพมหานครของหน่วยงานเอกชน

8. ข้อมูลของเส้นทางบางเส้นทางในบริเวณถนนเยาวราช ได้ใช้ข้อมูลทุติยภูมิที่รวบรวมโดยกองวิศวกรรมจราจร กรมทางหลวง รวมทั้งแขวงการทางกรุงเทพ กรมทางหลวงด้วยเช่นกัน

1.5 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงานแบ่งออกเป็น ขั้นตอนดังต่อไปนี้

1. กำหนดขอบเขตของการศึกษา และจุดประสงค์ต่างๆ
2. ศึกษาค้นคว้าเพิ่มเติมจากงานวิจัยต่างๆที่เกี่ยวข้องรวมทั้งบทความทางวิชาการซึ่งมีเนื้อหาเกี่ยวข้องกัน
3. ค้นคว้าเพิ่มเติมเกี่ยวกับหลักการที่จะนำไปใช้ในการวิเคราะห์และค้นหาเส้นทางตามที่ต้องการ
4. เก็บรวบรวมข้อมูลต่างๆที่จำเป็นต้องใช้ในการวิเคราะห์ ซึ่งเป็นการออกไปรวบรวมข้อมูลเองและเป็นข้อมูลทางทุติยภูมิที่ได้มีการสำรวจและรวบรวมไว้แล้ว โดยหน่วยงานราชการและเอกชน และนำมาเก็บเป็นฐานข้อมูลที่ใช้ในอนาคตต่อไป
5. ค้นคว้าเพิ่มเติมเกี่ยวกับหลักการในการเขียนโปรแกรมที่ใช้ในการค้นหาเส้นทางต่างๆ ตลอดจนโปรแกรมและอุปกรณ์ที่ใช้ในการเก็บภาพแผนที่ว่าควรจะใช้โปรแกรมอะไร อุปกรณ์อะไร เพราะอะไร และหาความแตกต่างของการใช้วิธีในการค้นหาเส้นทางที่แตกต่างกันแล้วสามารถสรุปได้ว่าวิธีที่ใช้มาทั้งหมดวิธีไหนเหมาะกับงานมากกว่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทำการเก็บภาพแผนที่เข้าเครื่องคอมพิวเตอร์ โดยใช้แผนที่มาตราส่วน 1 : 1000
7. เขียนโปรแกรมค้นหาเส้นทางด้วยภาษา Prolog เพื่อใช้ในการเปรียบเทียบการทำงานที่เหมาะสมที่สุดกับการศึกษานี้ โดยได้มีการกำหนดจุดต้นและจุดปลายของเส้นทาง เพื่อใช้ในการหาเส้นทางที่ใช้เวลาน้อยที่สุดในการเดินทาง
8. ทดสอบความถูกต้องของโปรแกรมที่เขียนขึ้นโดยการจำลองสถานการณ์ต่างๆที่จะเกิดขึ้นในการเดินทางได้ ตลอดจนหาความแตกต่างของแต่ละโปรแกรมว่าวิธีใดเหมาะสมที่สุด
9. สรุปผลการวิเคราะห์ข้อมูล ปัญหา และข้อเสนอแนะต่างๆ รวมถึงแนวทางในการแก้ปัญหาได้

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้หลักการทางด้านฐานความรู้ มาประยุกต์ใช้กับงานด้านเส้นทางการจราจรได้
2. เรียนรู้วิธีการใช้โปรแกรมทางด้าน CAD/CAM ต่างๆที่เป็นประโยชน์มากในการเก็บภาพแผนที่การจราจรบริเวณถนนเยาวราช
3. รู้จักวิธีการเชื่อมโยงผลจากโปรแกรมทางด้านปัญญาประดิษฐ์ มาใช้ในโปรแกรมสำเร็จรูปทางด้าน CAD/CAM ได้
4. รับทราบ เข้าใจกฎเกณฑ์และข้อจำกัดต่างๆที่นำมาประยุกต์ใช้เป็นข้อมูลในการตัดสินใจในการเลือกใช้อุปกรณ์ โปรแกรม ต่างๆ ให้เหมาะสมกับงานด้านเส้นทางการจราจร

1.7 ปัญหาต่างๆและแนวทางการแก้ปัญหา

1. ในการศึกษานี้ได้ใช้ข้อมูลทุติยภูมิที่ได้มีหน่วยงานราชการและเอกชนต่างๆ สำรวจและรวบรวมไว้แล้ว แต่เนื่องจากยังไม่มีหน่วยงานใด ที่มีข้อมูลตรงกับจุดประสงค์และการดำเนินงานของการศึกษานี้ แม้ว่าผู้จัดทำได้ทำการออกสำรวจเก็บข้อมูลที่ต้องการเองก็ตาม แต่เนื่องจากขีดความสามารถและปัจจัยต่างๆไม่ค่อยจะเอื้ออำนวยเท่าที่ควรทำให้การรวบรวมข้อมูลเป็นไปได้อย่างยากลำบากมาก จึงต้องอาศัยเวลาในการรวบรวมข้อมูลมาก

2. เนื่องจากการศึกษานี้จำเป็นต้องใช้ข้อมูลในการทำงานต่างๆเป็นจำนวนมาก แต่เนื่องจากปัญหาความล่าช้าในระบบราชการทำให้เกิดความไม่สะดวกและความล่าช้าในการทำงานมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ในเรื่องอัตราเร็วของรถในการเดินทางเป็นเวลาโดยประมาณที่เกิดขึ้นจากผู้จัดทำออกไปสำรวจและเก็บข้อมูลเองนั้น ทำให้ได้ข้อมูลที่ไม่ครอบคลุมทุกช่วงเวลา และข้อมูลที่ได้ก็ไม่สามารถใช้ได้กับทุกสถานการณ์ได้ เช่น ในกรณีซ่อมถนน รถชน เทศกาลต่างๆโดยเฉพาะเทศกาลของคนจีนที่ไม่การกำหนดวันเป็นสากลแน่นอนแต่จะถือตามปฏิทินของชาวจีน เป็นต้น จึงสมมติให้ใช้อัตราเร็วในช่วงการเดินรถในจราจรติดขัดเป็นหลัก

4. ในเรื่องข้อมูลตำแหน่งในแผนที่ และระยะทางในแผนที่ ถึงแม้จะใช้แผนที่ที่ได้มาตรฐานแล้วก็ตาม แต่ขนาดของแผนที่แต่ละแผ่นมีขนาดใหญ่มากและไม่เป็นมาตรฐานขนาดกระดาษสากล(คือใหญ่กว่าขนาดกระดาษ A0 อีก) ทำให้เวลาถ่ายเอกสารมาเพื่อใช้ในการคัดลอกผ่านทางเครื่องคัดลอก (Digitizer) ลงเครื่องคอมพิวเตอร์ ต้องทำออกเป็นแผ่นย่อยๆ ทำให้เกิดปัญหาบริเวณขอบของรอยต่อระหว่างแผ่นที่ไม่สนิทกันได้ ทำให้บางครั้งจึงมีความจำเป็นต้องสมมติรอยต่อเองบ้าง

5. โปรแกรมแต่เดิ มที่ใช้เป็นโปรแกรมภาษา LISP ที่อยู่ใน AutoCad Release12 for DOS ที่ชื่อว่า AutoLisp นั้นมีปัญหาคือไม่สามารถใช้งานได้กับภาษา LISP ทุกคำสั่งทำให้มีปัญหาในการใช้งานมากเมื่อนำโปรแกรมภาษา LISP ที่เขียนและสามารถใช้งานได้ใน LISP มาตรฐานมาใช้ก็ไม่สามารถรวมเข้ากันได้เพราะบางคำสั่งนั้นไม่สามารถทำงานได้ใน AutoLisp และเนื่องจากตัว AutoCad เองก็ไม่สามารถใช้ตัวอักษรภาษาไทยได้(ไม่ว่าจะใช้โปรแกรมภาษาไทยต่างๆ เช่น VTHAI.COM, TSM.EXE, THAIPRO.EXE เป็นต้น) จึงจำเป็นที่จะต้องแบ่งตัวโปรแกรมออกเป็นสองส่วน คือ ส่วนของการทำงานหาเส้นทางต่างๆโดยใช้โปรแกรมภาษา Prolog ที่สามารถใช้โปรแกรมภาษาไทยต่างๆได้และเป็นระบบฐานความรู้แบบหนึ่ง และส่วนของการแสดงผลที่รับผลจากโปรแกรมภาษา Prolog ดังกล่าวมาแสดงผลเป็นเส้นทางต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

ปัญหาต่างๆในปัจจุบันที่เกิดขึ้นนั้นสามารถแยกประเภทตามลักษณะการแก้ไขได้ 2 ประเภทคือ การแก้ไขโดยใช้โครงสร้างการแก้ปัญหาที่ตายตัวแน่นอนที่รับประกันว่าได้ผลลัพธ์แน่นอน เรียกว่า การคำนวณตามสูตร (Computation) แต่การแก้ปัญหาโดยใช้วิธีนี้จะสามารถจัดการกับปัญหาได้ บางประเภทเท่านั้น เช่น ปัญหาทางคณิตศาสตร์ เป็นต้น วิธีการทำก็ง่าย โดยการแปลงวิธีการต่างๆ ไปเป็นรูปอัลกอริทึม (Algorithm) ที่คอมพิวเตอร์สามารถทำงานตามได้ แต่ปัญหาที่เกิดขึ้นจริงๆในโลกมีน้อยมากที่สามารถแก้โดยใช้วิธีนี้ ส่วนใหญ่จะใช้แบบที่ 2 ทั้งสิ้น ซึ่งปัญหาแบบที่2 นี้จะประกอบด้วยปัญหาที่สามารถแก้ไขโดยการค้นหา (Search) ผลลัพธ์โดยจะต้องใช้ขบวนการทางปัญญาประดิษฐ์ (Artificial Intelligent) เป็นหลัก

2.1 เทคนิคในการค้นหา (Search Technique)

การค้นหามีอยู่หลายวิธี แต่มีวิธีที่นิยมและใช้กันมากๆ คือ

- การค้นหาตามแนวลึกก่อน (Depth-First Search : DFS)
- การค้นหาตามแนวกว้างก่อน (Breadth-First Search : BFS)
- การค้นหาแบบปีนตามเนินเขา (Hill-Climbing Search)
- การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด (Least-Cost Search)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 การประเมินค่าการค้นหาแต่ละวิธี (Evaluating a Search)

การประเมินค่าความสามารถของเทคนิคการค้นหารเป็นสิ่งที่ซับซ้อนมาก แต่สามารถใช้วิธีวัดที่สำคัญอยู่ 2 วิธีคือ

1. ดูว่าการค้นหาสามารถพบผลลัพธ์ใดๆได้เร็วแค่ไหน
2. การค้นหาสามารถพบผลลัพธ์ที่ดีใดๆได้เร็วแค่ไหน

มีปัญหาอยู่หลายประเภทที่การค้นหาผลลัพธ์หนึ่งใดๆ จะขึ้นอยู่กับใช้เวลาให้น้อยที่สุด (ใช้ความพยายามในการหาน้อยที่สุด) ดังนั้นก็ควรใช้ข้อ1 เป็นสำคัญ แต่ถ้าเป็นกรณีอื่น สิ่งสำคัญที่ต้องการคือ ผลลัพธ์ที่ใกล้เคียงกับผลลัพธ์ที่ถูกใจที่สุด (Optimal) ซึ่งความยาวของเส้นทาง(Path)ผลลัพธ์ และ จำนวนโนด (Node) ที่มีอยู่ก็เป็นตัวตัดสินความเร็วในการค้นหา แต่จำไว้ว่าการย้อนรอย (Backtracking) จากจุดตาย (Dead End) คือความพยายามที่สูญเปล่าในการค้นหา ดังนั้นต้องพยายามให้เกิดการย้อนรอยให้น้อยที่สุด

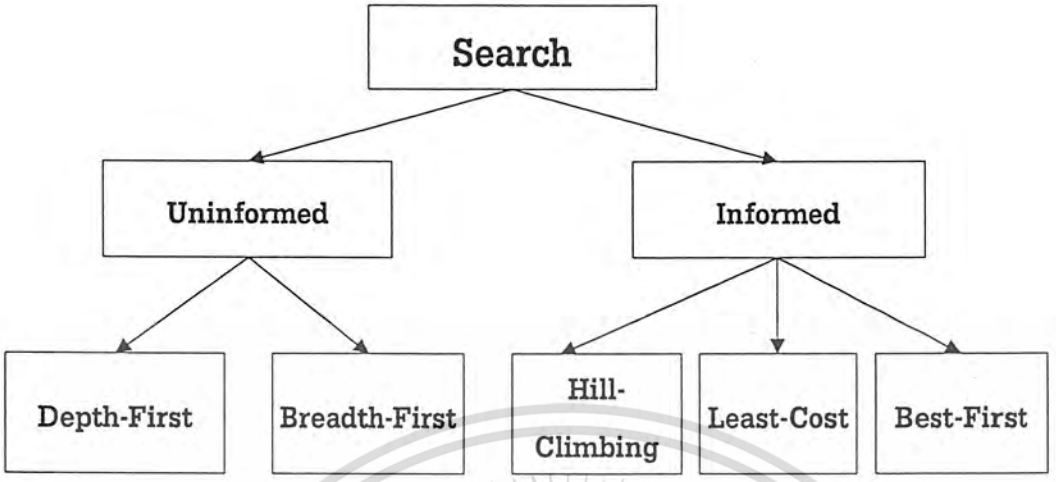
สิ่งสำคัญที่ต้องทำความเข้าใจคือความแตกต่างระหว่างการพบผลลัพธ์ที่ถูกใจ และการพบผลลัพธ์ที่ดี โดยที่การพบผลลัพธ์ที่ถูกใจมักจะต้องใช้ความพยายามในการค้นหาที่ค่อนข้างจะมากและอาจต้องใช้เวลานาน แต่การพบผลลัพธ์ที่ดีหมายถึง การพบผลลัพธ์ที่อยู่ภายในข้อจำกัดชุดหนึ่ง ซึ่งอาจจะไม่ใช่ผลลัพธ์ที่ดีที่สุดก็ได้

เทคนิคในการค้นหาแบ่งได้เป็น 2 ประเภทใหญ่ๆ (ดังรูป 2.1) คือ

1. **Uninformed State Space Search** เช่น การค้นหาแบบตามแนวลึกก่อน การค้นหาแบบตามแนวกว้างก่อน เป็นต้น การค้นหาประเภทนี้จะไม่มีการใช้ Heuristic มาช่วยเป็นข้อมูลในการค้นหาต่อไป

2. **Informed State Space Search** เช่น การค้นหาแบบปีนตามเนินเขา การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด การค้นหาเส้นทางที่ดีที่สุด (Best-First Search) เป็นต้น ซึ่งการค้นหาแบบนี้จะนำ Heuristic ต่างๆมาช่วยประกอบในการตัดสินใจค้นหาแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.1 แสดงประเภทของการค้นหาทั้ง 2 ประเภท

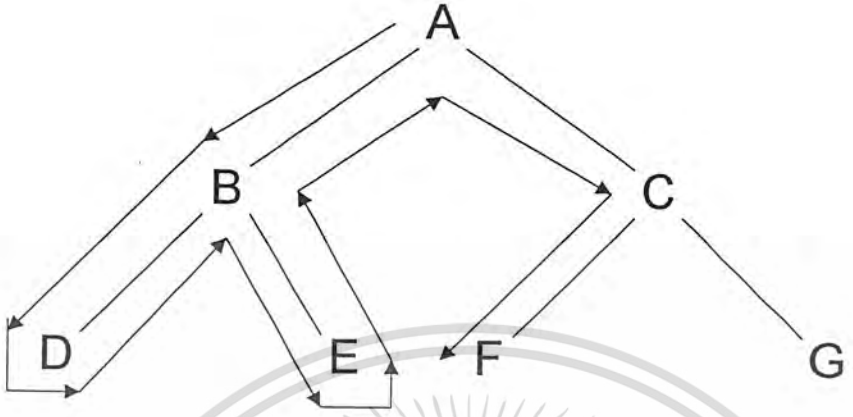
2.1.2 Uninformed State Space Search

2.1.2.1 การค้นหาแบบตามแนวลึกก่อน (Depth-First Search : DFS)

การค้นหาแบบตามแนวลึกก่อน หมายถึง การสำรวจเพื่อหาข้อสรุปทางเดินที่เป็นไปได้แต่ละทางที่ไปยังจุดหมาย (Goal) ก่อนที่จะสำรวจเส้นทางอื่นต่อๆไป เพื่อให้เข้าใจลองมาดูรูปต่อไปนี้ สมมติให้ F เป็นจุดหมาย ดังนั้นจากรูป 2.2 เราจะได้เส้นทางการค้นหาตามวิธีนี้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034938



รูป 2.2 แสดงตัวอย่างทางเดินในการค้นหาแบบตามแนวลึกก่อน

โดยมีลำดับในการค้นหาเป็น ABDBEBACF (การค้นหาแบบนี้จะเหมือนกับการค้นหาแผนภาพต้นไม้แบบตามลำดับ (Inorder Tree Traversal)) การค้นหาแบบนี้จะเคลื่อนไปยังด้านซ้ายมือเรื่อยๆจนกว่าจะถึงจุดสิ้นสุด แล้วก็จะย้อนกลับขึ้นมา 1 ระดับ เพื่อไปทางด้านขวามือ แล้วก็เคลื่อนไปทางด้านซ้ายเรื่อยๆจนกว่าจะพบจุดหมายหรือจุดสิ้นสุดใดๆ การทำงานนี้จะทำซ้ำไปเรื่อยๆจนกว่าจะพบจุดหมาย หรือได้ทำการตรวจสอบครบทุกจุดในทุกเส้นทางแล้ว

อัลกอริทึมของการค้นหาตามแนวลึกก่อน :

1. ถ้าสถานะเริ่มต้น (Initial State) เป็นสถานะจุดหมาย (Goal State) ก็ให้ออกจากโปรแกรมและผ่านค่าว่าทำงานสำเร็จแล้ว
2. นอกจากนั้นให้ทำตามขั้นตอนต่อไปนี้จนกว่าจะทำงานสำเร็จหรือล้มเหลว (Failure)
   a) ให้สร้างตัวสืบทอด (Successor) ของสถานะเริ่มต้นเป็น X ซึ่งถ้าไม่มีตัวสืบทอดอีกก็แสดงว่าการทำงานค้นหาล้มเหลว
   b) ทำการค้นหาอีกครั้งโดยใช้ X เป็นสถานะเริ่มต้น
   c) ถ้าทำงานสำเร็จก็ผ่านค่าว่าสำเร็จ ถ้าเป็นกรณีอื่นก็กลับไปที่ลูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.2 แสดงเส้นทางในการค้นหา จะเห็นว่าการค้นหาตามแนวลึกก่อนเป็นการหาที่จะพบจุดหมายแน่นอน (ถ้ามีจุดหมายอยู่ในเส้นทางใดเส้นทางหนึ่ง) แต่ทว่าอาจจะเป็นการค้นหาที่เหนื่อยที่สุด (Exhaustive Search) ซึ่งจากรูป 2.2 จะเกิดกรณีนี้ได้ถ้า G เป็นจุดหมายที่ต้องการ การค้นหาแบบตามแนวลึกก่อนนี้เป็นวิธีที่ดีที่สุดในการค้นหาเส้นทางแบบไม่มี Heuristic (Uninformed State Space Search)

การประเมินค่าของการค้นหาตามแนวลึกก่อน

จากตัวอย่างข้างต้น จะเห็นได้ว่าการค้นหาตามแนวลึกก่อนจะเป็นการค้นหาที่ดีมาก ถ้าสามารถหาพบได้ในครั้งแรกๆ โดยไม่ต้องมีการย้อนรอยกลับ แต่ก็จะไม่ค่อยดีนัก ถ้าต้องสำรวจเกือบทุกจุดเพื่อหาผลลัพธ์ที่ถูกใจที่สุด อย่างไรก็ตามสังเกตว่าการค้นหาแบบตามลึกก่อนจะเป็นสิ่งที่แย่มากถ้าต้องสำรวจไปตามเส้นทางยาวๆ แล้วพบว่าไม่มีจุดหมายที่ต้องการอยู่เลยแม้ว่าจะไปถึงจุดสุดท้ายแล้ว ซึ่งถ้าเป็นกรณีนี้จะทำให้เสียเวลามาก เพราะไม่เพียงจะต้องสำรวจไปตามเส้นทางแล้วยังต้องทำการย้อนรอยกลับไปอีกเพื่อไปหาจุดหมายที่ต้องการ ซึ่งถ้าเป็นกรณีนี้ก็ควรจะใช้วิธีค้นหาแบบตามแนวกว้างก่อน

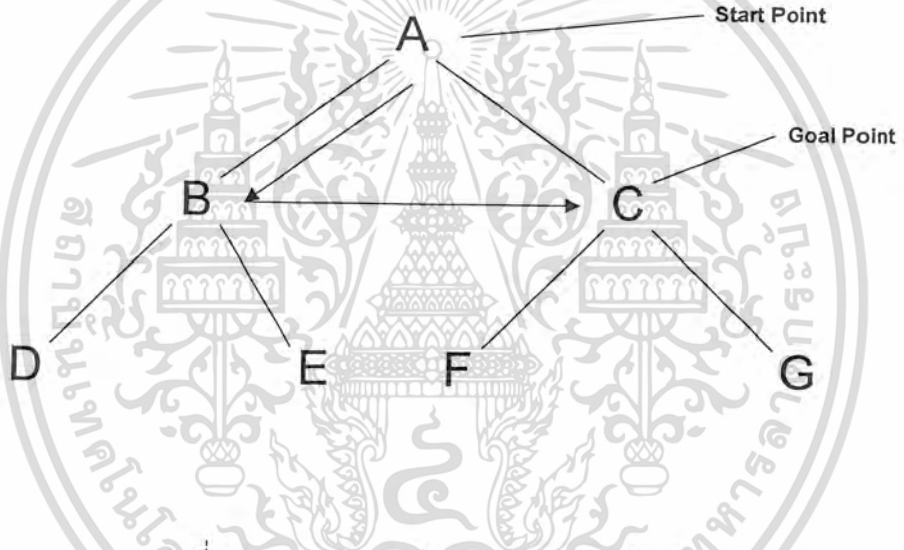
ข้อดีของการค้นหาแบบตามแนวลึกก่อน

- การค้นหาแบบนี้ใช้เนื้อที่หน่วยความจำน้อย เพราะว่าจะใช้เก็บเฉพาะจุด (Node) ที่อยู่ในเส้นทางปัจจุบันเท่านั้น ซึ่งตรงข้ามกับการค้นหาแบบตามแนวกว้างก่อน ที่จะต้องเก็บเส้นทางทุกเส้นทางในแผนภาพต้นไม้ ก่อนที่จะทำการสร้างเส้นทางต่อไป
- ถ้าจะพูดถึงโอกาสที่จะพบจุดหมายแล้ว (หรือถ้ามีการเรียงลำดับสถานะต่อไปอย่างเหมาะสมแล้ว) การค้นหาแบบตามแนวลึกก่อนอาจจะพบผลลัพธ์โดยไม่ต้องตรวจสอบจุดต่างๆ มากมายนัก ซึ่งตรงข้ามกับการค้นหาแบบตามแนวกว้างก่อนที่ต้องทำการสำรวจทุกส่วนในแผนภาพต้นไม้ทั้งต้น โดยสำรวจที่ระดับ n ก่อนแล้วค่อยไปที่ n+1 ไปเรื่อยๆ จนกว่าจะพบหรือไม่พบจุดหมายที่ต้องการเลย ซึ่งนี่เป็นสิ่งสำคัญถ้ามีผลลัพธ์ที่ต้องการอยู่เป็นจำนวนมากแล้ว วิธีการค้นหาแบบตามแนวลึกก่อนสามารถหยุดเมื่อพบผลลัพธ์อันหนึ่งอันใดแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.2 การค้นหาแบบตามแนวกว้างก่อน (Breadth-First Search)

การค้นหาแบบตามแนวกว้างก่อนจะตรงข้ามกับการค้นหาแบบตามแนวลึกก่อน โดยวิธีนี้จะต้องตรวจสอบแต่ละจุด (Node) ที่อยู่ในระดับเดียวกัน ก่อนที่จะไปยังระดับที่ลึกลงไป ดังรูป 2.3 จะเห็นว่าถ้าสมมติให้ C เป็นจุดหมายที่ต้องการ เส้นทางในการทำงานจะเป็นดังนี้ ABC วิธีการนี้เหมือนกับวิธีการค้นหาตามแนวลึกก่อน ตรงที่รับประกันได้ว่าสามารถหาผลลัพธ์ได้แน่นอน (ถ้ามีจุดหมายที่ต้องการในเส้นทาง) แต่อาจจะต้องเป็นการค้นหาที่เหนื่อยที่สุด (Exhaustive Search)



รูปที่ 2.3 แสดงภาพการค้นหาแบบตามแนวกว้างก่อน

อัลกอริทึมของการค้นหาแบบตามแนวกว้างก่อน :

1. สร้างตัวแปรขึ้นมาตัวหนึ่ง เรียกว่า NODE-LIST และตั้งให้เป็นสถานะเริ่มต้น
2. ทำขั้นตอนต่อไปนี้จนกว่าจะถึงสถานะจุดหมาย หรือค่าใน NODE-LIST ว่าง
   a) เอาค่าตัวแรกสุดใน NODE-LIST ออกและเรียกมันว่า X ซึ่งถ้าเอาออกแล้ว NODE-LIST ว่าง ก็ให้ออกจากการทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b) สำหรับแต่ละทางที่แต่ละกฎสามารถสอดคล้อง (match) กับสถานะที่ได้อธิบายใน X แล้วให้ทำต่อไปนี้

i) ประยุกต์กฎเพื่อที่จะสร้างสถานะใหม่

ii) ถ้าสถานะใหม่เป็นสถานะจุดหมาย ให้ออกการทำงานและผ่านค่าสถานะนี้ออกไป

iii) นอกจากนี้ให้เพิ่มสถานะใหม่ไปต่อท้ายของ NODE-LIST

การประเมินค่าการค้นหาแบบตามแนวกว้างก่อน

จากตัวอย่างข้างต้นนั้นจะเห็นว่า การค้นหาระบบตามแนวกว้างก่อนจะทำงานได้ดีโดยการพบผลลัพธ์แรกที่ไม่ต้องมีการย้อนรอย และผลลัพธ์ที่ได้นี้จะเป็นผลลัพธ์ที่ถูกใจที่สุด ซึ่งตามความเป็นจริง ผลลัพธ์ 3 อันแรกเป็นเส้นทางที่ดีที่สุด 3 ทางอยู่แล้ว อย่างไรก็ตามขอให้จำไว้ว่า ผลลัพธ์ที่ได้จะไม่สามารถใช้ได้ทั่วไปกับสถานการณ์อื่นๆ เพราะว่ามันขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูลที่อยู่ในคอมพิวเตอร์

ข้อดีของการค้นหาแบบตามแนวกว้างก่อน

- การค้นหาแบบตามแนวกว้างก่อนจะไม่หลงกล (trapped) อยู่กับการสำรวจที่ขาดสิ่งนำทาง (blind) ซึ่งตรงข้ามกับการค้นหาตามแนวลึกก่อนที่อาจจะค้นหาตามเส้นทางที่ไม่ถูกต้องอยู่แล้วเพียงเส้นเดียวเป็นระยะเวลานาน หรือบางที่อาจจะตลอดไป ก่อนที่เส้นทางดังกล่าวจะหยุดที่ไม่มีตัวสืบทอด (non-successor) ซึ่งกรณีนี้จะเป็นปัญหามากกับการค้นหาแบบตามแนวกว้างก่อน ถ้ามีลูปเกิดขึ้น (เช่น สถานะที่ตัวสืบทอดก็คือ หนึ่งในตัวสร้าง (Ancestor) เอง เป็นต้น) นอกจากจะมีการสร้างการทำงานพิเศษรองรับส่วนนี้
- การค้นหาแบบตามแนวกว้างก่อนนี้จะรับประกันว่า จะพบผลลัพธ์ถ้ามีผลลัพธ์อยู่จริง ซึ่งถ้ามีผลลัพธ์อยู่มากมายแล้ว ผลลัพธ์ที่ใช้เส้นทางน้อยที่สุด (Minimal) จะถูกพบ (เช่น ใช้ขั้นตอนในการเดินทางน้อยที่สุด เป็นต้น) ซึ่งถ้า สิ่งนี้เกิดขึ้นด้วยความจร
กว่าจะไม่ได้การสำรวจจนกว่าเส้นทางที่สั้นกว่าอื่นๆ จะได้รับการสำรวจแล้ว ซึ่งตรงข้ามกับการค้นหาแบบตามแนวลึกก่อน ที่อาจจะพบผลลัพธ์เป็นเส้นทางที่ยาวที่สุดก็ได้ โดยที่ไม่ได้มีการสำรวจเส้นทางอื่นๆที่สั้นกว่าเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 Informed State Space Search (Heuristic Search)

ตอนนี้ทั้งการค้นหาตามแนวลึกก่อน และการค้นหาตามแนวกว้างก่อน เป็นการค้นหาแบบขาดสิ่งนำทาง (Blind Search) โดยทั้ง 2 วิธีจะทำการหาผลลัพธ์โดยเดาการเคลื่อนจากจุดหมายหนึ่งไปยังจุดอื่นโดยปราศจากการเดาที่มีหลักการ ซึ่งนี่อาจเป็นสิ่งที่ดีสำหรับบางสถานการณ์ที่ควบคุมได้ถ้ามีข้อมูลที่เพียงพอที่จะบอกได้ว่าควรใช้วิธีไหนถึงจะดีกว่า แต่โปรแกรมทาง AI โดยทั่วไปแล้วต้องการข้อมูลที่น้อยกว่านั้นก็ทำงานได้ดีกว่าเทคนิคทั้ง 2 วิธีนั้น และวิธีที่ใช้กันโดยการค้นหาที่เพิ่มความสามารถทาง Heuristic เข้ามาร่วมด้วย

ในการแก้ปัญหาที่ยากจำนวนมากอย่างมีประสิทธิภาพ จำเป็นต้องมีการตกลงประนีประนอมเกี่ยวกับความต้องการของระบบและการเปลี่ยนแปลงที่จะเกิดขึ้น เพื่อจะสามารถสร้างโครงสร้างการควบคุมที่ไม่รับประกันว่าจะหาคำตอบได้ แต่จะหาคำตอบที่ดีมากๆให้เสมอ ซึ่งวิธีการนี้จำเป็นต้องใช้ Heuristic ดังนั้นควรมาทำความเข้าใจความหมายของคำว่า Heuristic ก่อน ดังนี้

Heuristic คือ เทคนิคอันหนึ่งที่พัฒนาประสิทธิภาพของขบวนการค้นหาอันหนึ่ง ซึ่งจะเป็นไปได้ว่าคำตอบที่ได้จะไม่ใช่คำตอบที่ดีที่สุดก็ตาม Heuristic ก็เหมือนเป็นไกด์นำเที่ยว ซึ่งมีข้อดีคือ จะชี้จุดที่มักเป็นที่สนใจของคนทั่วไป แต่อาจจะพลาดในจุดที่เป็นที่สนใจเฉพาะบางคน โดยจะช่วยให้เป็นไกด์ของขบวนการค้นหาที่ไม่ต้องการความสมบูรณ์ 100% เหมือนวิธีอื่นๆ แต่โดยเฉลี่ยแล้ว วิธีนี้จะพัฒนาคุณภาพของเส้นทางที่สำรวจเสมอ ถ้าเราใช้ Heuristic ที่ดีก็สามารถหวังได้ว่าจะได้รับคำตอบที่ดีในการแก้ปัญหายากได้ เช่น ปัญหาของการเดินทางของคนขายของ (Traveling Salesman) ซึ่งใช้เวลาในการค้นหาน้อยกว่า exponential และมี Heuristic ที่สามารถทำงานได้ทั่วไปที่ดีบางตัวที่เป็นประโยชน์ในการแก้ปัญหาหลักๆบางตัวที่เกิดขึ้นอย่างแพร่หลายมากมาย แต่ก็สามารถสร้าง Heuristic พิเศษที่สามารถแก้ไขปัญหาเฉพาะได้ ตัวอย่างเช่น ลองสมมติว่าคุณเดินหลงทางในป่า และต้องการน้ำซึ่งป่าเป็นป่าที่ลึกมากและต้นไม้ก็ใหญ่มากเกินกว่าจะปีนขึ้นไปดูสภาพรอบๆได้ อย่างไรก็ตามคุณมีความรู้อยู่ 4 อย่างคือ

1. แม่น้ำ ลำธาร สระ มักจะอยู่ในหุบเขา
2. สัตว์ต่างๆ มักจะเดินทางไปสู่แหล่งน้ำ
3. เมื่อคุณใกล้แม่น้ำคุณจะได้กลิ่นน้ำ
4. คุณสามารถได้ยินเสียงน้ำไหล

ดังนั้นในการหาน้ำคุณเริ่มต้นโดยการเดินลงจากเขา เพราะว่าน้ำจะไม่อยู่บนยอดเขา ต่อไปเมื่อคุณพบรอยเท้ากวางที่วิ่งลงจากเขา คุณก็ตามไปเพราะว่าคุณรู้ว่ามันจะนำไปหาแหล่งน้ำ และเมื่อคุณได้ยินเสียงน้ำไหลเบาๆ คุณก็รู้ว่าน้ำอยู่ทางนี้ คุณก็ไปตามเสียงที่ได้ยิน และเมื่อคุณได้รับรู้ถึงความชื้นที่เพิ่มขึ้นในอากาศ คุณสามารถได้กลิ่นน้ำ สุดท้ายคุณก็พบสายน้ำได้และได้ดื่มมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังตัวอย่างนี้แสดงให้เห็นว่าข้อมูลทาง Heuristic ถึงแม้ว่าจะไม่ถูกต้องและไม่รับประกันว่าถูกต้องหรือไม่ แต่ก็เพิ่มโอกาสให้สามารถค้นหาไปยังจุดหมายได้เร็วขึ้น หรือถูกใจมากขึ้น (Optimal) หรือจะพูดอีกอย่างว่า มันเพิ่มโอกาสที่จะสำเร็จได้เร็วขึ้น

ในเรื่องของ Heuristic นั้น นอกจากจะมีการค้นหาแบบ Heuristic แล้ว ยังมีอีกสิ่งหนึ่งที่สำคัญ คือ ฮิวริสติก ฟังก์ชั่น (Heuristic Function) ซึ่งหมายถึง ฟังก์ชั่นที่ทำหน้าที่ในการวัดขนาดของความเป็นไปได้ในการแก้ปัญหาซึ่งจะแสดงด้วยตัวเลข โดยวิธีการดังกล่าวจะกระทำโดยการพิจารณาถึงวิธีการ (aspects) ต่างๆที่ใช้ในการแก้ปัญหา ณ สถานะหนึ่งว่าจะสามารถแก้ปัญหาได้ตามที่ต้องการหรือไม่ โดยกำหนดเป็นน้ำหนักที่ให้กับการแก้ปัญหาของแต่ละวิธี น้ำหนักเหล่านี้จะถูกแสดงด้วยตัวเลขที่กำกับไว้กับโหนดต่างๆ ในกระบวนการค้นหาและค่าเหล่านี้จะเป็นตัวที่ใช้ในการประมาณความเป็นไปได้ว่า เส้นทางที่ผ่านโหนดนั้นจะมีความเป็นไปได้ในการนำไปสู่หนทางการแก้ปัญหาได้มากน้อยแค่ไหน

จุดประสงค์ที่แท้จริงของ Heuristic function คือ การกำกับทิศทางของกระบวนการค้นหา เพื่อให้อยู่ในทิศทางที่ได้ประโยชน์มากที่สุด โดยการบอกว่าเราควรเลือกเดินเส้นทางไหนก่อน ในกรณีที่มีเส้นทางมากกว่าหนึ่งเส้นทางต้องเลือก

2.1.3.1 ลักษณะของปัญหาที่ควรใช้ Heuristic

การค้นหาแบบ Heuristic เป็นวิธีทั่วไปวิธีหนึ่งที่ใช้ในการแก้ปัญหาได้จำนวนมาก และยังสามารถเขียนเป็นกรณีพิเศษเพื่อใช้กับปัญหาเฉพาะด้านได้อย่างดีอีกด้วย ซึ่งวิธีการเลือกวิธีที่เหมาะสมที่สุด (หรือเอาหลายวิธีมารวมกัน) สำหรับปัญหาเฉพาะด้านก็มีความจำเป็นต้องวิเคราะห์ปัญหาออกเป็นมิติต่างๆกันหลายมิติดังนี้

- ปัญหาสามารถแยกเป็นปัญหาย่อยๆเล็กๆและง่ายๆได้ไหม
- ขั้นตอนในการหาผลลัพธ์จะสามารถถูกทิ้งไปหรือไม่ทำต่อไปได้ไหม ถ้ารู้ว่านี่ เป็นวิธีการที่ไม่ฉลาดนัก
- สามารถรู้ของเขตของปัญหาได้หรือไม่
- ผลลัพธ์ที่ดีของปัญหาสามารถรู้เองได้โดยไม่ต้องนำไปเปรียบเทียบกับผลลัพธ์อื่นๆ ก็รู้ว่าดีแน่นอนได้หรือไม่
- ผลลัพธ์ที่ต้องการเป็นสถานะของงานหรือเส้นทางไปยังสถานะนั้นๆ
- ต้องการความรู้จำนวนมากในการแก้ปัญหาหรือต้องการความรู้ที่สำคัญเท่านั้นก็เพียงพอต่อการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอมพิวเตอร์สามารถให้ผลลัพธ์ออกมาเมื่อเราให้ปัญหาเข้าไปได้เอง หรือจำเป็นต้องทำการติดต่อระหว่างมนุษย์กับคอมพิวเตอร์ถึงจะได้คำตอบ

วิธีการค้นหาทาง Heuristic จำนวนมากมักจะยึดตามแง่มุมของปัญหาที่เกิดมากที่สุดหรือน้อยที่สุด ซึ่งในที่นี้จะใช้ Heuristic 2 ตัวในการตรวจสอบโดยที่ทั้ง 2 ตัวนั้นตรงข้ามกันและให้ผลต่างกันด้วย

2.1.3.2 การค้นหาแบบปีนตามเนินเขา (Hill-Climbing Search)

ถ้าสมมติว่าเราต้องเดินทางจาก A ไป F ในรูป 2.3 และมีข้อจำกัดในการเดินทางอยู่ 2 ข้อคือ

1. จำนวนของจุดที่ต้องเดินทางต้องน้อยที่สุด
2. ความยาวของเส้นทางต้องน้อยที่สุด (สังเกตว่า เส้นทางที่สั้นที่สุดไม่ได้หมายความว่า จะมีจุดที่ผ่านน้อยที่สุด)

อัลกอริทึมที่ใช้ในการค้นหาที่พยายามหาผลลัพธ์อันแรกที่มีจำนวนจุดที่ผ่านน้อยที่สุด จะใช้ข้อมูลทาง Heuristic ที่แสดงถึงระยะทางที่ผ่านที่ยาวขึ้น แต่ใกล้จุดหมายมากขึ้น โดยการลดจุดที่ต้องผ่านโดยใช้วิธีทาง AI นั้น ต้องใช้การค้นหาแบบที่ใช้ข้อมูลทาง Heuristic ที่เรียกว่า **การค้นหาแบบปีนตามเนินเขา (Hill-Climbing)**

โดยปกติแล้วอัลกอริทึมของการค้นหาแบบปีนตามเนินเขานั้น จะทำการเลือกขั้นตอนต่อไปเพื่อที่จะได้อยู่ที่จุดที่ใกล้จุดหมายมากที่สุด ซึ่งชื่อของอัลกอริทึมนี้ได้รับจากการอุปมานว่านักเดินทางไกลที่หลงอยู่ในความมืดของเวลากลางคืน และยืนอยู่บนครึ่งทางของภูเขา ถ้าสมมติว่าที่พักของนักเดินทางอยู่บนยอดภูเขา แม้จะอยู่ในความมืด นักเดินทางก็รู้ว่าแต่ละก้าวที่เดินไปนั้นไปถูกทางแล้ว

สำหรับการเดินทางจากจุด A ไป F นั้น จำเป็นต้องรู้ข้อมูลบางอย่างที่จะช่วยนำทางในการเดินทางได้ ซึ่งก็คือ Heuristic ของโปรแกรมการเดินทาง เช่น ระยะทางระหว่างจุด ค่าน้ำหนัก (Weight) ระหว่างจุด เป็นต้น โดยเริ่มจากการเลือกเส้นทางการเดินทางที่อยู่ไกลที่สุดจากจุดที่เราอยู่ ปัจจุบันโดยหวังว่าวิธีนี้จะทำให้อยู่ใกล้จุดหมายมากขึ้น ซึ่งผลที่ได้จะค่อนข้างดี คือจะมีจำนวนจุดที่ผ่านน้อยที่สุด ซึ่งใกล้เคียงกับการเดินทางที่สั้นที่สุด (Shortest route) และไม่ค่อยจะมาเสียเวลาในการย้อนรอยกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมของการค้นหาแบบปีนตามเนินเขา :

1. สร้างคำตอบในขั้นแรกขึ้นมาแล้วดูว่าเป็นคำตอบสุดท้ายได้หรือไม่ ถ้าได้ก็เลิก ถ้าไม่ได้ให้ทำต่อข้อถัดไป
2. จากคำตอบนี้ ให้ใช้กฎทั้งหลายหาคำตอบใหม่ออกมา แล้วให้เป็นคำตอบขั้นแรกชุดใหม่ขึ้นมา
3. ในแต่ละองค์ประกอบ (element) ของข้อ 2 ให้ทำดังนี้
   3.1 ตรวจสอบด้วยฟังก์ชั่นตรวจสอบ ถ้าเป็นคำตอบ ให้ยกเลิก
   3.2 ถ้าไม่ใช่ ให้ตรวจสอบว่าเป็นตัวใดที่ใกล้เคียงที่สุด

   เมื่อเทียบคำตอบกับค่าที่ได้ออกมาทุกตัวที่ทดสอบมาแล้วว่าเป็นคำตอบได้หรือไม่ ถ้าได้ให้จำคำตอบไว้ ถ้าไม่ได้ยกเลิก
4. ใช้องค์ประกอบที่ดีที่สุดจากข้อ 3 และให้เป็นคำตอบสมมติของตัวต่อไป ในขั้นตอนนี้ จะสัมพันธ์กับการแก้ปัญหาในห้วงปัญหาตามทิศทางที่ปรากฏซึ่งจะนำไปสู่คำตอบได้เร็วที่สุด
5. กลับไปที่ขั้นตอนที่ 2

การประเมินค่าวิธีของการค้นหาแบบปีนตามเนินเขา

การค้นหาแบบปีนตามเนินเขา ค่อนข้างดีในหลายๆสถานการณ์ เพราะว่ามันตั้งใจจะลดจำนวนของจุดที่ต้องผ่านก่อนที่จะถึงจุดหมาย แต่อย่างไรก็ตามก็มักจะเกิดปัญหาอยู่ 3 กรณีคือ

1. ***Foothill Problem (Local Maximum)*** เกิดในการณีที่ผลลัพธ์ที่ได้จะดีที่สุดเฉพาะบริเวณนี้แต่ไม่ใช่ดีที่สุดของทั้งหมด หมายถึง เป็นส่วนที่บอกสถานะที่ดีที่สุดเมื่อเทียบกับสถานะข้างเคียงเท่านั้น แต่ถ้าเทียบกับสถานะอื่นๆ ที่อยู่ห่างออกไปหรือการตรวจสอบขั้นต่อไปแล้ว ไม่แน่ว่าจะได้ผลออกมาดีที่สุด การแก้ปัญหาเช่นนี้ทำได้โดยอาศัยกระบวนการของการย้อนรอย (backtracking)
2. ***Ridge Problem*** มักจะเกิดจากอัลกอริทึมที่เป็นสาเหตุให้เกิดสันเขาที่ต้องข้ามตลอดเวลาซึ่งสันเขาที่ว่านี้เป็นสิ่งที่สูญเปล่า ซึ่งเป็นผลที่เกิดจากการที่ฟังก์ชั่นทดสอบตรวจสอบออกมาว่าออปชั่นที่กำลังทดสอบนี้มีคะแนนดีกว่าออปชั่นที่อยู่ในระดับถัดไป ซึ่งแสดงว่าโหนดลูกที่ถูกสร้างขึ้นมาใหม่ที่ควรจะเข้าใกล้คำตอบมากกว่า แต่ผลกลับบอกไม่ใช่ ปัญหาเช่นนี้สามารถแก้ไขได้โดยการใช้กฎมาอธิบายปรากฏการณ์นี้ก่อนที่จะทำการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ***Plateau Problem*** เป็นเหมือนการวิ่งอยู่บนสนามฟุตบอลที่มีเสาโทรศัพท์ตั้งอยู่เป็นจุดๆ ซึ่งจะทำให้การทำงานโดยวิธีที่ใช้ Heuristic นี้จะไม่มีการพัฒนาอะไรเลย เป็นเหมือนกับการใช้วิธีการค้นหาแบบตามแนวลึกก่อน นั่นคือ ในกรณีที่ปัญหาอยู่บนระดับเดียวกัน และผลของการหาค่าจาก Heuristic function ที่ได้เท่ากันหมด จะไม่สามารถตัดสินใจได้ว่า จะเลือกเส้นทางเส้นไหน ซึ่งเรื่องนี้เราสามารถแก้ได้ด้วยก้าวกระโดด

2.1.3.3 การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด (Least-Cost Search)

เป็นวิธีการที่ตรงข้ามกับการค้นหาแบบปีนตามเนินเขา โดยมีแนวคิดที่คล้ายกับการยืนอยู่ตรงกลางถนนบนเนินเขาใหญ่แห่งหนึ่งขณะที่ใส่รองเท้าสเก็ต (skate) อยู่ ซึ่งจะรู้สึกได้ว่าการที่จะไหลลงไปมันง่ายกว่าจะเลื่อนขึ้นมา ดังนั้นก็คือการใช้ค่าใช้จ่ายให้น้อยที่สุดในการหาเส้นทางที่สามารถเดินทางได้โดยสะดวก คือเดินทางได้โดยใช้ความพยายามน้อยที่สุด (Least-Cost takes the path of least effort)

การประยุกต์วิธีการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุดนี้กับการเดินทางจากจุดหนึ่งไปอีกจุดหนึ่งเสมือนว่าเป็นเส้นทางเชื่อมต่อที่สั้นที่สุดที่จะเกิดขึ้นในกรณีทั้งหมด ดังนั้นเส้นทางที่พบจะมีโอกาสที่จะเป็นเส้นทางที่ระยะทางสั้นที่สุด ซึ่งต่างกับวิธีการค้นหาแบบปีนตามเนินเขา ที่จะมีจำนวนจุดที่ผ่านน้อยที่สุด แต่วิธีการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด จะมีจำนวนระยะทางที่ใช้น้อยที่สุด

การประเมินค่าของวิธีการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด

วิธีนี้มีข้อดีเป็นข้อเสีย และข้อเสียเป็นข้อดีของวิธีการค้นหาแบบปีนตามเนินเขา โดยวิธีนี้สามารถแก้ปัญหาของวิธีการค้นหาแบบปีนตามเนินเขาได้ทั้ง 3 ข้อแต่ก็เกิดปัญหาด้าน ผลลัพธ์ที่ได้อาจไม่ถูกต้องนัก (False Valley), Low Land, และทางแคบ (Gorge) แต่ไม่ได้หมายความว่าวิธีนี้จะดีกว่าวิธีการค้นหาแบบปีนตามเนินเขา เพราะวิธีไหนจะดีกว่าก็ขึ้นอยู่กับกรณีของปัญหาที่พบ โดยแต่ละวิธีก็มีข้อดีของแต่ละอย่าง แต่ทั้ง 2 วิธีเป็นการค้นหาที่ไม่ใช้การค้นหาแบบไม่มีสิ่งนำทาง (non-blind search)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การเลือกว่าควรจะใช้วิธีการค้นหาแบบไหน

ดังที่เห็นว่าเทคนิคด้าน Heuristic นั้น ทำงานได้ดีกว่า แต่ก็ไม่ได้หมายความว่าจะต้องใช้ Heuristic เพราะว่าบางทีถ้ามีข้อมูลไม่เพียงพอในการตัดสินว่าควรจะไปยังจุดไหนต่อไปหรือเพื่อไปยังเส้นทางที่ไปหาจุดหมาย ซึ่งเราสามารถแยกกฎสำหรับการเลือกวิธีในการค้นหาออกเป็น 2 กรณีคือ

1. สำหรับปัญหาที่สามารถใช้ประโยชน์ได้จากการค้นหาแบบ Heuristic และ
2. สำหรับปัญหาที่ไม่สามารถใช้ประโยชน์ได้จากการค้นหาแบบ Heuristic

ถ้าสามารถประยุกต์ Heuristic เข้ากับปัญหาได้ สิ่งที่ได้จะดีกว่า แต่ข้อยกเว้นอย่างหนึ่งของกฎนี้คือสถานการณ์ที่มีความรู้ที่บอกว่า การค้นหาแบบตามแนวกว้างดีกว่า เช่น เมื่อฐานข้อมูลถูกจัดในแบบที่ปรากฏแน่ชัดว่าการค้นหาแบบตามแนวกว้างทำงานได้ดีกว่าแน่นอน

ส่วนการเลือกว่าควรใช้วิธีการค้นหาแบบ Heuristic ทั้งแบบปีนตามเนินเขาและแบบเสียค่าใช้จ่ายน้อยที่สุดนั้น ขึ้นอยู่กับข้อจำกัดที่ว่า ต้องการพยายามให้ผลลัพธ์ที่ได้เป็นแบบมากที่สุด (maximal) หรือน้อยที่สุด (minimal) ซึ่งโดยทั่วไปแล้วการค้นหาแบบปีนตามเนินเขาจะสร้างผลลัพธ์ด้วย จำนวนจุดที่ผ่านที่น้อยที่สุด ในขณะที่การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด จะสร้างเส้นทางที่ต้องใช้ความพยายามในการให้น้อยที่สุด

2.2 ส่วนการวาดภาพแผนที่โดยใช้เครื่อง Digitizer

การวาดแผนที่ลงในเครื่องคอมพิวเตอร์นั้นมีโปรแกรมต่างๆที่ใช้ในการสนับสนุนมากมาย โปรแกรมที่สร้างมาเพื่อสนับสนุนนั้นมีอยู่ประเภทหนึ่งที่นิยมใช้กันมากในการวาดรูปต่างๆ คือ CAD/CAM เพราะมีอุปกรณ์ที่คอยช่วยสนับสนุนอยู่มากมาย และโปรแกรมหนึ่งที่ใช้ในโปรแกรม AutoCAD RELEASE 12 FOR DOS เหตุผลที่ใช้เนื่องจากว่า จากการศึกษามาพบว่าจุดประสงค์หลักอันหนึ่งของโปรแกรมนี้คือ สามารถใช้ในการวาดแผนที่ ดังนั้นจึงมีอุปกรณ์สนับสนุนมากมาย และยังสามารถเชื่อมต่อกับอุปกรณ์ภายนอก เช่น เมาส์ Digitizer Plotter เป็นต้น ได้เป็นอย่างดี และจากการศึกษาพบอีกว่า Digitizer น่าจะเป็นอุปกรณ์ที่เหมาะสมที่สุดแล้วในการคัดลอกแผนที่จากแผ่นกระดาษให้ปรากฏในจอภาพคอมพิวเตอร์ เพราะนอกจากจะช่วยลดเวลาในการคัดลอกแล้ว ยังสามารถปรับปรุงแก้ไขได้ในภายหลังโดยที่มาตราส่วนไม่ผิดเพี้ยนไป ซึ่งตามความเป็นจริงแล้วการใช้เมาส์ก็สามารถทำหน้าที่นี้ได้ แต่วิธีการและขั้นตอนในการเก็บภาพจะเป็นไปอย่างยากลำบากช้า และค่อนข้างยุ่งยากในการเก็บ ดังนั้นก็พอสามารถสรุปข้อดีที่ Digitizer ได้เปรียบเมาส์มีอยู่หลายประการ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Digitizer จะสามารถกำหนดโคออร์ดิเนต และจำค่าโคออร์ดิเนตที่ผ่านมาได้ แม้แต่จะยกตัวชี้ไปยังตำแหน่งอื่นๆบนกระดานDigitizerก็ตาม โดยจะมีความสัมพันธ์กันในทุกๆจุดบนDigitizer แต่เมาส์จะไม่สามารถอ้างอิงถึงตำแหน่งเดิมหรือตำแหน่งใดๆ ที่เมาส์เคลื่อนที่ไปได้เลย โดยถือว่าทุกๆจุดที่มีการเคลื่อนไหวคือ จุดกำเนิดใหม่ทุกครั้ง เพราะฉะนั้นDigitizer จึงเหมาะที่จะใช้ลอกแบบหรือแผนที่ได้อย่างมีประสิทธิภาพ โดยเฉพาะเส้น Contour ในแผนที่

2. Digitizer เมื่อใช้คู่กับคำสั่ง Tablet จะมีความสามารถในการใช้ Tablet Menu ในการเลือกคำสั่งต่างๆ ภายใน Tablet Menu ออกมาใช้งานโดยที่เรามิต้องป้อนคำสั่งใดๆ ทาง Command Line แต่เมาส์จะไม่มีคุณสมบัติทางด้านนี้เลย

2.2.1 ทำไมต้อง Digitizer

ตามความเป็นจริงแล้วการนำภาพแผนที่ลงไปยังเครื่องคอมพิวเตอร์นั้นยังมีวิธีที่ง่ายกว่านี้และสะดวกกว่านี้อีก คือ การ Scan โดยใช้เครื่อง SCANNER แต่วิธีการ scan นี้เป็นวิธีการที่นำรูปภาพทั้งรูปเข้าไปเก็บในลักษณะของรูปภาพ แต่สำหรับหัวข้อในการทำการศึกษาเส้นทางจราจรนี้จำเป็นต้องมีการแก้ไขส่วนต่างๆของภาพได้ แต่ถ้าเป็นลักษณะของการ scan แล้วการแก้ไขจะเป็นไปอย่างยากกว่าและไม่สามารถนำมาแก้ทีละส่วนได้อย่างมีประสิทธิภาพในภายหลัง และการใช้วิธีการ scan ภาพจะต้องใช้เนื้อที่ในการเก็บภาพเป็นจำนวนมากกว่าการ Digitize มากนัก ด้วยเหตุผลดังกล่าวพอจะสรุปเหตุผลที่ต้องใช้เครื่อง Digitizer ไม่ใช้เครื่อง Scanner ดังนี้

1. การ scan ภาพทำได้อย่างรวดเร็วกว่าการ Digitize แต่การนำส่วนเล็กๆของภาพมาแก้ไขในภายหลังเป็นไปได้อย่างยากมากกว่า และในหัวข้อการศึกษาเส้นทางจราจรนั้นจำเป็นต้องมีการเปลี่ยนแปลงค่อนข้างบ่อยดังนั้นการ Digitize จึงเป็นสิ่งที่ควรทำมากกว่า

2. ภาพที่เก็บโดยการ scan จะกินเนื้อที่มากกว่าภาพที่เก็บโดยการ Digitize เพราะว่าการเก็บภาพด้วยวิธีการ scan จะเป็นการเก็บภาพในทุกจุดของภาพโดยไม่สนใจว่าที่จุดนั้นมีภาพอยู่หรือไม่ก็ตาม แต่การ Digitize จะเก็บภาพเฉพาะเส้นที่เป็นภาพเท่านั้นเป็นลักษณะการเก็บภาพแบบ Vector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนการเขียนโปรแกรม AutoLISP

AutoLisp ถือว่าเป็นภาษาคอมพิวเตอร์ภาษาหนึ่ง ที่ผู้ใช้จะสามารถสั่งรัน (run) ขณะทำงานบน AutoCAD ซึ่งจะช่วยให้ AutoCAD มีความฉลาดขึ้นเท่าที่ต้องการ ถือได้ว่าเป็นการนำประสิทธิภาพสูงสุดของ AutoCAD มาใช้งาน และโดยปกติแล้ว AutoCAD จะถูกสร้างขึ้นมาเพื่อใช้ได้กับงานทั่วๆไปเท่านั้น จะไม่สามารถใช้ได้กับงานเฉพาะทางได้อย่างคล่องตัว แต่ AutoLISP นี้จะช่วยดัดแปลงให้ AutoCAD สามารถนำมาใช้ได้กับงานเฉพาะทางได้อย่างไม่มีขีดจำกัด

2.3.1 อะไรคือ AutoLISP

AutoLISP แท้ที่จริงแล้วเป็นภาษาคอมพิวเตอร์ที่ถูกสร้างขึ้นมาโดยอาศัยพื้นฐานมาจากภาษา LISP (LISt Processing) ที่สามารถเขียนขึ้นในลักษณะที่เป็นโครงสร้าง (Structure) ซึ่งจะเอื้ออำนวยให้ผู้ใช้งานสามารถเขียนแยกส่วน หรือแบ่งกันทำงานได้ แต่ AutoLISP นี้ถือว่าเป็นตัวแปลภาษาที่จะต้องถูกเรียกใช้งานภายใต้ AutoCAD เท่านั้น จะไม่สามารถถูกเรียกใช้งานตามลำพัง โดยจะถูกคอมไพล์ในลักษณะอินเตอร์เพรเตอร์ (Interpreter) คือเมื่อป้อนข้อมูลลงไปก็จะให้คำตอบออกมาทีละบรรทัดต่อบรรทัด ทันที

AutoLISP จะใช้งานร่วมกับมาโครของ AutoCAD ในการทำงานและเราจะใช้ AutoLISP นี้เป็นเครื่องมือในการสร้าง Command ขึ้นมาใหม่ด้วยการนำคำสั่งต่างๆของ AutoCAD ที่มีอยู่แล้วมาประกอบกัน ใช้ฟังก์ชั่นพิเศษเข้ามาช่วยเสริม และใส่เงื่อนไขต่างๆ เข้าไป เพื่อให้สามารถตัดสินใจเองได้ จากนั้นก็จะนำทั้งหมดนี้มาประมวลเข้าด้วยกัน สร้างเป็น Command ขึ้นมาใหม่ ที่จะสามารถเรียกใช้ได้เหมือนกับคำสั่งแบบทั่วๆไปของ AutoCAD

2.3.2 โครงสร้างของ AutoLISP

AutoLISP นี้นับได้ว่าเป็นภาษาโครงสร้างภาษาหนึ่งที่มีลักษณะการเขียนโปรแกรมคล้ายกับภาษา C หรือ Pascal แต่จะกระทำได้ง่ายกว่าตรงที่ว่าไม่ต้องมีประกาศ หรือกำหนดชนิดของตัวแปรขณะเริ่มต้นโปรแกรม ซึ่งจะทำให้เขียนได้ง่ายเหมือนภาษา BASIC นับได้ว่าเป็นผลดีต่อการเขียนโปรแกรม AutoLISP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะอาศัยหลักการเป็นภาษาโครงสร้างนี้ มาใช้ให้เป็นประโยชน์ โดยเราทำการแบ่งโปรแกรมออกเป็นส่วนๆ ที่แยกกันทำงานเป็นแบบโมดูล (module) ซึ่งจะง่ายต่อการเขียนโปรแกรมและตรวจหาที่ผิด (Debug) นอกจากนี้ยังสามารถนำโมดูลนี้ไปใช้ได้กับโปรแกรมตัวอื่นๆได้ ซึ่งถือได้ว่าเป็นการเขียนโปรแกรมแบบมีประสิทธิภาพมากที่สุด

ดังนั้นเราควรจะมีการวางแผนการเขียนโปรแกรมไว้ล่วงหน้าก่อนที่จะเขียนโปรแกรมเสมอ โดยการวิเคราะห์และเขียนโฟลว์ชาร์ตขึ้นมา จากนั้นจึงพยายามทำการรวมคำสั่งที่ต้องถูกเรียกใช้งานบ่อยๆ หรือกลุ่มคำสั่งที่ซับซ้อนมากๆ เข้าด้วยกัน และควรเขียนโปรแกรมขึ้นในลักษณะแบบบล็อก (Block) ที่เป็นฟังก์ชั่นที่สามารถถูกเรียกใช้งานได้ แล้วค่อยเขียนโปรแกรมเมนูหลักให้ทำการดึงฟังก์ชั่นนั้นมาใช้งานอีกทีภายหลัง ดังตัวอย่าง LISP ต่อไปนี้

```
(defun *error* (Msg)
        (setvar "cmdecho" cmd "blipmode" cblp)
        (princ "Error : ")
        (princ Msg)
        (terpri)
        (princ))
; จบส่วนโมดูลแรก และจะใช้ *error* (Msg) เป็นตัวที่ถูกเรียกใช้งานได้
(defun default (prompt def)
        (setq Ucode (getvar "lunits"))
        (setq Uprec (getvar "luprec"))
(setq prompt "<<" (rtos def ucode uprec) ">> : "))
; จบส่วนโมดูลที่สองที่เป็นฟังก์ชั่นที่สามารถถูกเรียกใช้งานได้จาก Main โปรแกรม
(defun C: Ptos(s)                    ; โปรแกรม Main หลัก
        .....
        .....
        .....
(setq texth (getdist (default  "Text Height : "  tex)))
        .....
        .....
; จบส่วน Main โปรแกรม
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่าง LISP ที่ผ่านมา คงจะทำให้เรามองลักษณะการเขียนโปรแกรมแบบแยกเป็นโมดูลที่นับว่าเป็นประโยชน์ต่อการพัฒนาโปรแกรม AutoLISP ได้อย่างมีประสิทธิภาพ

2.3.3 องค์ประกอบของโปรแกรม AutoLISP

ในการเขียนโปรแกรม AutoLISP ขึ้นมาใช้งานนั้น จำเป็นจะต้องมีส่วนประกอบที่สำคัญดังนี้

1. ค่าคงที่และตัวแปร (Constants and Variables)
2. ฟังก์ชั่นและอาร์กิวเมนต์ (Function and Arguments)
3. เครื่องหมายดำเนินการ (Operations)
4. นิพจน์ (Expressions)

ยกตัวอย่างเช่น

```
(setq X (+ M 2))
```

เมื่อ + เป็นฟังก์ชั่น
M เป็นตัวแปร
2 เป็นค่าคงที่

โดยทั้งหมดนี้ถือว่าเป็น 1 นิพจน์ ดังมีรายละเอียดดังต่อไปนี้

ค่าคงที่และตัวแปร (Constant and Variables)

1. ***ค่าคงที*** จะหมายถึง ค่าที่มีค่าคงที่ตลอดเวลา โดยไม่มีการเปลี่ยนแปลง และจะคงที่เสมอตราบที่ยังดำเนินการประมวลผลอยู่ เช่น 5, 4, 2 ฯลฯ

2. ***ตัวแปร*** จะหมายถึง สัญลักษณ์ที่ผู้เขียนสร้างขึ้นเอง เพื่อใช้ในการเก็บข้อมูลต่างๆ ในขณะทำการประมวลผล และค่าของตัวแปรนี้จะมีการเปลี่ยนแปลงได้ตลอดเวลาขณะมีการประมวลผล ซึ่งการกำหนดสัญลักษณ์จะต้องเป็นไปตามกฎเกณฑ์การตั้งตัวแปร นอกจากนี้ตัวแปรยังแบ่งออกได้ 6 ชนิด คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 *เลขจำนวนเต็ม (Integer)* ตัวแปรชนิดนี้จะเป็นตัวเลขที่เก็บค่าจำนวนเต็มเท่านั้น ซึ่งจะหมายถึง ค่าบวก ลบ และค่าศูนย์ เช่น 1, -1, 0, 2 ฯลฯ แต่ก็ยังมีข้อจำกัดของการเก็บตัวแปรอยู่ที่ว่าจะเก็บค่าได้ในช่วง -32768 ถึง +32768 เท่านั้น

2.2 *เลขจำนวนจริง (Real)* ตัวแปรชนิดนี้จะเก็บค่าจำนวนจริงที่สามารถเก็บค่าทศนิยมได้ด้วย ทำให้ไม่จำกัดขนาดในการเก็บ เช่น 2.4E-8, 1.4E15, 0.25 แต่ถ้าในกรณีที่ตัวเลขมีค่าน้อยกว่า 1 จะต้องใส่เลขศูนย์นำหน้าเสมอ เช่น 0.25, 0.92 ฯลฯ

2.3 *รายการ (List)* จะเป็นตัวแปรที่เก็บค่าตัวแปรชุดหรือเป็นกลุ่มเซต และข้อมูลของตัวแปร list นี้จะต้องอยู่ภายใต้เครื่องหมายวงเล็บเปิดและปิดเสมอ ซึ่งจะเหมาะสมกับการเก็บตัวแปรที่มีความสัมพันธ์ในลักษณะเป็นกลุ่มหรือเซต เช่นการเก็บค่าตำแหน่งโคออร์ดิเนต (X, Y, Z) โดยจะต้องใช้การเก็บค่าตัวแปรแบบ list ดังนี้

(setq Pd (list 2.5, 11,9)) ก็จะได้

Pd = (2.5, 11, 9)

2.4 *ข้อความ (String)* จะเป็นตัวแปรที่ใช้ในการเก็บอักขระหรือข้อความต่างๆ ซึ่งจะต้องถูกกำหนดไว้ภายใต้เครื่องหมายคำพูด " " เช่น

(setq Text "EMPIRE")

2.5 *ตัวระบุไฟล์ (File Descriptor)* จะใช้เป็นตัวแปรในการเก็บหมายเลขของไฟล์ที่เรากำหนด เพื่อที่ว่าจะสามารถอ้างอิงนำมาใช้งานได้เมื่อต้องการ โดยเฉพาะในกรณีที่เราจะกระทำการต่างๆที่เกี่ยวกับไฟล์ เช่นในการอ่านหรือเขียนไฟล์ที่เราจะต้องกำหนดไฟล์ไว้ก่อนเช่น

(setq filepd "TEST.DOC")

(setq fis (open filepd "r"))

เราจะได้ fis เป็นตัวแปรแบบ File Descriptor ในการเปิดไฟล์ข เพื่อทำการอ่านเข้ามา

2.6 *ชื่อของวัตถุ (Entity Name)* จะเป็นตัวแปรในการเก็บชื่อของ Object หรือ Entity ซึ่งใน AutoCAD จะทำการเก็บข้อมูลทุกชนิดในรูป Database แบบ list และเรายังที่จะสามารถดึงนำมาใช้งานได้ เช่น

```
((-1 <Entity name 60000230>)        ; ถูกกำหนดชื่อในรูปรหัส
( 0     LINE)                       ; เป็นเส้นตรง
( 8     "0")                        ; อยู่เลเยอร์ 0
( 39    3.00000)                    ; ความหนาเท่ากับ 3
.....
.....
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 *Selection Sets* จะเป็นตัวแปรที่เกิดจากการเลือก Object ในขณะที่มีพร็อมต์ปรากฏขึ้นให้เราทำการเลือก

Select Objects :

โดยขณะเราทำการเลือกออบเจ็กต์ AutoCAD จะทำการเก็บกลุ่มที่เลือกไว้ใน Selection Sets ที่เราจะสามารถอ้างอิงได้ และนำมาใช้โดยการเรียกผ่านฟังก์ชั่นของ AutoLISP

ฟังก์ชั่นและอาร์กิวเมนต์ (Funciton and Arguments)

ฟังก์ชั่น จะถือว่าเป็นตัวดำเนินการ (operate) ทำให้เกิดผลลัพธ์ขึ้นมาใหม่ โดยการใช้ค่าคงที่หรือตัวแปรเข้ามาประมวลผลผ่าน operate ในลักษณะต่างๆ เช่น +, -, *, / ฯลฯ

อาร์กิวเมนต์ แท้ที่จริงก็คือค่าคงที่หรือตัวแปรที่เราส่งเข้าไปดำเนินการ (Operate) ด้วยฟังก์ชั่นต่างๆ นั่นเอง เช่น

(setq x (/ 2 y))

เมื่อ / เป็น ฟังก์ชั่น

2 y เป็น อาร์กิวเมนต์

เครื่องหมายดำเนินการ (Operator)

ใน AutoCAD ก็จะมีเครื่องหมายดำเนินการที่เหมือนกับภาษาคอมพิวเตอร์ที่เป็นภาษาอื่น ซึ่งถือว่าเป็นสิ่งที่จำเป็นอย่างมากที่จะทำให้ AutoLISP มีการตัดสินใจได้เอง ซึ่งพอจะแบ่งออกได้ 2 แบบ ดังนี้

1. *เครื่องหมายเปรียบเทียบ หรือเรียกว่า Relational Operator* โดยเมื่อทำการเปรียบเทียบแล้วก็จะแสดงว่า เป็นจริง T (True) หรือเป็นเท็จ nil (False) อย่างใดอย่างหนึ่ง เช่น ให้ A = 4 และ B = 5 แล้วให้

- (setq R1 (< A B)) จะได้ค่า R1 = T เพราะ B > A จริง
- (setq R2 (> A B)) จะได้ค่า R2 = nil เพราะ A > B ไม่จริง

ซึ่งพอจะสรุปได้เป็นตารางที่ 2.1 ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องหมายดำเนินการ	ความหมาย
>	มากกว่า
<	น้อยกว่า
<=	น้อยกว่า หรือ เท่ากับ
>=	มากกว่า หรือ เท่ากับ
=	เท่ากับ
/=	ไม่เท่ากับ
eq	เท่ากันอย่างแท้จริง
equal	เท่ากับ (ใช้กับตัวแปรแบบ list)
listP	ตัวแปรแบบ list หรือไม่
minUSP	จำนวนมีค่าลบหรือไม่
numberP	ตัวแปรแบบ Real หรือ Integer หรือไม่
zeroP	ตัวแปรเป็นศูนย์หรือไม่

ตารางที่ 2.1 แสดงเครื่องหมายดำเนินการและความหมาย

2. *เครื่องหมายทางตรรกศาสตร์ หรือเรียกว่า Logical Operator* ซึ่งจะประกอบไปด้วยลักษณะ 3 ประการคือ

2.1 AND ซึ่งจะให้ค่า T ก็ต่อเมื่อมีทุกๆสถานการณ์เป็น T หมด

2.2 OR จะใช้เปรียบเทียบสถานะแบบ เลือกค่าใดค่าหนึ่ง ซึ่งมีเพียงค่าใดค่าหนึ่งเป็น T ก็จะมีค่ารวมเป็น T

2.3 NOT จะใช้ในการกลับเปลี่ยนสถานะ ให้มีลักษณะที่ตรงกันข้ามกับสถานะเดิม

นิพจน์ (Expression)

นิพจน์จะเป็นกลุ่มของชุดคำสั่ง ที่นำฟังก์ชั่นอาร์กิวเมนต์ หรือเครื่องหมาย ดำเนินการต่างๆ เข้ามาประมวลผลเข้าด้วยกันให้ได้ผลลัพธ์ใหม่ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(setq M (+ A 12))

เมื่อ + เป็น ฟังก์ชั่น

A 12 เป็น อาร์กิวเมนต์

โดยที่ทั้งหมดเราเรียกว่า นิพจน์

ดังนั้นในโปรแกรม AutoLISP แต่ละโปรแกรมจะประกอบด้วยนิพจน์จำนวนหลายๆ นิพจน์ที่ประกอบกันขึ้นมา และการสร้างนิพจน์ขึ้นมาจะต้องเป็นไปตามกฎเกณฑ์การสร้างนิพจน์ดังนี้

1. ทุกๆนิพจน์ ไม่ว่าจะประกอบด้วยอาร์กิวเมนต์หรือไม่ก็ตาม จะต้องขึ้นต้นด้วยเครื่องหมายวงเล็บเปิดก่อน และจะต้องมีเครื่องหมายวงเล็บปิดตามท้าย เมื่อจบนิพจน์เสมอ

2. จำนวนของวงเล็บเปิดจะต้องเท่ากับวงเล็บปิดเสมอ ไม่ว่าจะซ้อนกันกี่นิพจน์ก็ตาม เช่น

(setq M (* 2 (+ x 3)))

เมื่อ + x 3 เป็น นิพจน์ตัวที่ 1

* 2 (+ x 3) เป็น นิพจน์ตัวที่ 2

setq M (* 2 (+ x 3))) เป็น นิพจน์ตัวที่ 3

3. ในกรณีที่มีนิพจน์ซ้อนกันมากกว่า 1 นิพจน์ AutoLISP จะทำการคำนวณวงเล็บในสุดก่อน แล้วจึงคำนวณวงเล็บนอกที่ถัดออกมาเรื่อยๆ จนจบนิพจน์กลุ่มนั้น

4. จะต้องมีช่องว่าง 1 อักษร (เคาะแป้น spacebar 1 ครั้ง) ระหว่างฟังก์ชั่น และอาร์กิวเมนต์เสมอ เช่น

(setq M (+ 2 5))

5. ในขณะที่เราให้นิพจน์ของเราทำการรับข้อมูลจากภายนอก เช่น จากเมาส์ หรือ คีย์บอร์ด หรืออุปกรณ์ช่วยอื่นๆ ถ้าเราไม่ป้อนค่าใดๆ ลงไปแล้วกด Enter ก็จะถือว่ามีค่าเป็น nil จนกว่าจะรับข้อมูลใหม่จึงจะมีค่าเป็น T

2.3.4 การเขียนและการใช้งาน AutoLISP

การเขียน AutoLISP ขึ้นมาใช้งานบน AutoCAD ถือได้ว่าเป็นการสร้างฟังก์ชั่นใหม่ขึ้นมานั่นเอง ซึ่งในการเขียน LISP ขึ้นมาใหม่นั้น จะต้องมีการกำหนดให้เป็นฟังก์ชั่นเสียก่อน ด้วยการ DEFUN ซึ่งจะย่อมาจาก Define FUNction ซึ่งจะมีลักษณะการกำหนดรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(DEFUN C:NAME (Arguments)
        (Expression)
        (Expression)
        .....
        .....
)
```

ซึ่งจะต้องมีวงเล็บเปิดและวงเล็บปิดที่สมดุลกันเสมอ ดังนั้นในการเขียนทุกครั้งควรสร้างวงเล็บขึ้นมาก่อนแล้วจึงเขียน Expression แทรกเข้าไป ซึ่งจะช่วยลดความผิดพลาดจากการเขียนโปรแกรมลงได้มาก

NAME นี้จะเป็นชื่อของฟังก์ชั่นที่เราจะสร้างขึ้นมา ซึ่งจะใช้อักษรตัวเล็กหรือใหญ่ก็ได้ แต่ไม่ควรซ้ำกับชื่อฟังก์ชั่นที่มีอยู่แล้ว

C : จะมีหรือไม่มีก็ได้ แต่ถ้าเราใส่นำหน้า NAME ก็จะหมายถึงเราจะสามารถเรียกใช้งานทาง Command line : ได้เลย แต่ถ้าเราไม่ใส่ C : ก็จะหมายถึงเราไม่สามารถเรียกใช้งานโดยทาง Command line : ได้ แต่จะต้องสร้างฟังก์ชั่นขึ้นมาเรียกใช้งานอีกทอดหนึ่ง

Argument จะเป็นสะพานเชื่อมไปยังฟังก์ชั่นตัวอื่น ซึ่งเราจะให้มีหรือไม่มีก็ได้ แต่ถ้าเราใส่ Argument ลงไปก็จะมีการส่งค่าไปยังฟังก์ชั่นอื่นได้ เหมาะสำหรับการแยกเขียนทีละโมดูล หลายๆโมดูล

Expression หรือนิพจน์จะเป็นรายการคำสั่งที่เราได้เขียนขึ้นมาให้ AutoCAD ได้ทำงานตามขั้นตอนต่างๆ ตามที่เราได้วางไว้

ในการเขียนโปรแกรม AutoLISP ขึ้นมานั้น เราจะต้องใช้โปรแกรมจำพวก Text Editor ในการสร้างโปรแกรมขึ้นมาในรูปของเท็กซ์ไฟล์ แล้วเราจะต้องใส่นามสกุล (File Type) เป็น .LSP เสมอ จากนั้นจึงทำการเรียกรันบน AutoCAD และโปรแกรมพวก Text Editor ที่เราเลือกมาใช้ในการเขียน AutoLISP นี้ควรเลือกใช้ให้เหมาะสม ซึ่งควรจะเป็นโปรแกรมที่มีขนาดเล็ก ใช้งานง่ายและไม่เสียเวลาในการติดต่อเรียกใช้มากนัก เพราะในทางปฏิบัติแล้วเราจะต้องทำการแก้ไขและกลับเข้าไปทดสอบรันโปรแกรม LISP อยู่ตลอดเวลาอย่างหลีกเลี่ยงไม่ได้

ดังนั้นเวลาจึงเป็นสิ่งสำคัญรองลงมา เพราะการกลับเข้าไปในโปรแกรม AutoCAD จะต้องใช้เวลาอยู่ช่วงหนึ่ง ถ้าเป็นไปได้ควรจะติดต่อผ่าน AutoCAD จะเป็นการดีที่สุด (ใช้ ACAD.PGP ในการติดต่อ) หรือใช้โปรแกรมจำพวก Resident (ฝังในหน่วยความจำ) ที่เป็น Text Editor ด้วยจะเป็นการเหมาะสมที่สุด และจากประสบการณ์ที่ใช้เอง คิดว่า Qeditor หรือ Sidekick น่าจะเหมาะสมที่สุดในขณะนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 การใช้งาน AutoLISP บน AutoCAD

หลังจากที่เราได้สร้างโปรแกรมและเก็บไฟล์ไว้ในรูป AutoLISP ที่นามสกุล .LSP เสร็จเรียบร้อยแล้ว จะต้องทำการเรียกโปรแกรมเข้ามาในหน่วยความจำก่อนเรียกการโหลดโปรแกรม ซึ่งจะมีรูปแบบดังนี้

(LOAD "file-name")

ซึ่งเราจะต้องใส่พาทให้ถูกต้องด้วย และ AutoCAD จะใช้เครื่องหมาย / แทนเครื่องหมาย \ จากที่เคยใช้งานปกติ เช่น ต้องการโหลดโปรแกรม TEST.LSP จาก B:\DATA\TEST.LSP ก็จะใช้รูปแบบคือ

```
Command : (load "B:/DATA/TEST")
```

หรือจะใช้เครื่องหมาย \\ แทนได้อีกด้วย เช่น

```
Command : (load "B:\\DATA\\TEST")
```

แต่ก็จะยังมี LISP พิเศษที่จะต้องถูกเรียกเข้ามาในหน่วยความจำอย่างอัตโนมัติทุกๆ ครั้งที่เข้าโปรแกรม AutoCAD นั่นคือ ACAD.LSP ซึ่งเราสามารถกำหนดฟังก์ชั่นต่างๆ ที่ต้องการให้โหลดเข้าสู่หน่วยความจำไว้ในไฟล์นี้ก่อน และเมื่อเข้าโปรแกรม AutoCAD ใหม่ทุกๆ ครั้ง โปรแกรมก็จะทำการค้นหาไฟล์ชื่อ ACAD.LSP เมื่อเจอก็จะทำการโหลดเข้าสู่หน่วยความจำอัตโนมัติทันที

นอกจากวิธีการโหลดด้วยการพิมพ์คำสั่งทาง Command Line แล้ว AutoCAD ยังจะ มีวิธีการโหลดที่สะดวกมากขึ้น นั่นก็คือการใช้เมนูในการโหลด ซึ่งจะเป็น Pull-Down Menu ในตัวเลือก Applications... ซึ่งในตัวเลือกนี้จะเกิดไดอะล็อกบ็อกซ์ขึ้น ให้เราทำการเลือกไฟล์ที่เป็นแอปพลิเคชั่น โดยจะเป็นพวกนามสกุล .LSP หรือ .EXP (นามสกุล .EXP จะเป็นแอปพลิเคชั่นจำพวก ADS) เข้ามาสู่ไดอะล็อกบ็อกซ์

นอกจากนี้เรายังสามารถทำการโหลดและไม่โหลด (Unload) (พวก ADS) ทีละ 1 ไฟล์ หรือเป็นกรุ๊ปก็ได้ หรือใช้การ pick ที่ File... ของ pull-down menu ที่อยู่ด้านบนของหน้าจอเพื่อไปเลือกแอปพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมที่มีหลายฟังก์ชั่นในไฟล์เดียวกัน

ในกรณีที่เราต้องการบรรจุฟังก์ชั่นที่ต้องการให้เรียกใช้งานหลายๆ ฟังก์ชั่นไว้ในไฟล์เดียวกันเช่น ACAD.LSP เราก็จะสามารถกระทำได้โดยการวางตำแหน่งของแต่ละฟังก์ชั่นไว้ในลักษณะเป็นบล็อกที่เรียงต่อกัน ดังนี้

```
(defun C: FUNC1 ()
        (Expression)
        (Expression)
        .....
        (Expression)
)                       ; จบฟังก์ชั่นที่ 1
(defun C: FUNC2 ()
        (Expression)
        (Expression)
        .....
        (Expression)
)                       ; จบฟังก์ชั่นที่ 2
```

แต่จะต้องใส่เครื่องหมายวงเล็บให้ครบตามแต่ละบล็อกอย่างถูกต้องด้วย เช่น

```
(defun C: P1 ()
        (Prompt "Call for function # 1")
        (Princ)
)                       ; จบฟังก์ชั่นที่ 1
(defun C: P2 ()
        (Prompt "Call for function #2")
        (Princ)
)                       ; จบฟังก์ชั่นที่ 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเรารันโปรแกรม LISP ตัวนี้บน AutoCAD เราก็จะได้คำสั่งใหม่เพิ่มขึ้นมาอีก คือ P1 และ P2 ที่สามารถเรียกใช้งานทาง Command Line ได้

ฟังก์ชั่นพื้นฐานใน AutoLISP

ฟังก์ชั่นที่ถูกสร้างขึ้นมาเพื่อใช้งานบน AutoLISP นั้นมีอยู่มากประมาณ 200 คำสั่ง ซึ่งแต่ละฟังก์ชั่นก็จะมีจุดประสงค์ของการใช้งานแตกต่างกันไป แต่ในกาใช้งานจริงๆ แล้วจะมีคำสั่งที่สำคัญในการใช้งานอยู่เพียงไม่มากนักที่จะถูกนำมาสร้างในโปรแกรม แต่อย่างไรก็ตามในที่นี้จะไม่อธิบายฟังก์ชั่นทั้งหมดแต่จะแบ่งออกเป็นประเภทการทำงานและยกตัวอย่างแค่บางฟังก์ชั่นเท่านั้น ซึ่งแยกเป็นประเภทได้ดังนี้

1. *ฟังก์ชั่นทางคณิตศาสตร์*
   เช่น +, -, *, /, REM, 1+, 1-, ABS เป็นต้น
2. *ฟังก์ชั่นที่เปลี่ยนแปลงชนิดของข้อมูล*
   เช่น ANGTOS (เปลี่ยนตัวเลขค่ามุมเป็นตัวอักษร)
   RTOS (เปลี่ยนจำนวนจริงให้เป็นสตริง)
   ATOI (เปลี่ยนตัวอักษรให้เป็นจำนวนเต็ม) เป็นต้น
3. *ฟังก์ชั่นที่เกี่ยวกับ String*
   เช่น STRCAT (เชื่อมต่อข้อความ)
   STRLEN (หาความยาวของสตริง) เป็นต้น
4. *ฟังก์ชั่นที่ควบคุมเกี่ยวกับการวนลูปและซับรูทีน*
   เช่น REPEAT (ทำซ้ำตามจำนวนที่กำหนด)
   WHILE (วนรอบจนกว่าเงื่อนไขจะเป็นเท็จ)
   IF (กำหนดเงื่อนไขให้นิพจน์ทำงาน) เป็นต้น
5. *ฟังก์ชั่นที่ควบคุมเกี่ยวกับวิธีการ List*
   เช่น ASSOC (ค้นหาค่าอะตอมใน LIST)
   CAR (นำค่าแรกใน LIST มาใช้)
   MEMBER (ค้นหาค่าอะตอมใน LIST) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. *ฟังก์ชั่นที่ควบคุมเกี่ยวกับการ Input & Output*

เช่น COMMAND (เรียกใช้คำสั่งใน AutoCAD)

GETSTRING (รับค่าที่เป็นข้อความ)

LOAD (เรียกไฟล์ของ AutoLISP เข้ามาเก็บไว้ในหน่วยความจำ)

เป็นต้น

7. *ฟังก์ชั่นที่เกี่ยวกับ Entity และการเลือก Entity*

เช่น SSGET (เลือก entities มาเก็บไว้ใน selection sets)

SSNAME (แสดงชื่อของ entity ใน selection sets)

ENTNEXT (แสดงชื่อ entities และรายการของ entities ใน database) เป็นต้น

8. *ฟังก์ชั่นที่สำคัญอื่นๆ*

เช่น NULL (ไม่มีค่า)

TRACE (ตรวจสอบโปรแกรม Debugging)

ERROR (แสดงข้อผิดพลาดของโปรแกรม) เป็นต้น

การใช้งานบนฟังก์ชั่นต่างๆ ของ AutoLISP

จากที่ผ่านมาจะเป็นการสรุปถึงความหมายของฟังก์ชั่นต่างๆบน AutoLISP อย่างพอสังเขปเท่านั้น แต่อย่างไรก็ตามการใช้งานในแต่ละคำสั่งนั้นจะใกล้เคียงกันไม่ซับซ้อนนัก ซึ่งอาจจะนำวิธีการใช้งานของอีกคำสั่งหนึ่งมาใช้งานได้ ดังนั้นในส่วนนี้จึงจะยกตัวอย่างการใช้งานฟังก์ชั่นเพียงเล็กน้อยเท่านั้นแต่คำสั่งอื่นๆก็มีวิธีการใช้งานที่คล้ายๆกันเช่นกัน

พื้นฐาน

ในการคอมไพล์โปรแกรม AutoLISP จะเป็นการคอมไพล์ในลักษณะอินเตอร์พริตเตอร์ (Interpreter) ที่จะมีการคำนวณและแปรคำสั่งในลักษณะบรรทัดต่อบรรทัด ซึ่งก็จะหมายความว่า AutoCAD จะทำการปฏิบัติตามคำสั่งทีละ 1 นิพจน์ ที่อยู่ในแต่ละบรรทัด ภายใต้วงเล็บเท่านั้น ซึ่งจะคล้ายกับเครื่องคิดเลขโดยจะให้ค่าออกมาทันที หลังจากที่เราป้อนค่าเข้าไปแล้ว

นอกจากการเขียนโปรแกรมขึ้นมาแล้ว เรายังอาศัยลักษณะที่เป็น interpreter นี้ติดต่อกับ AutoLISP โดยตรง ในขณะที่อยู่ command prompt ได้เลยหรือที่เราเรียกว่า Direct Mode เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Command : (+ 4 2)                          <นำค่า 4 มาบวกกับ 2 และจะให้ค่าเท่ากับ 6>
6
Command : (setq num (SQRT 3))              <ให้ num = รากที่สองของ 3 และมีค่าเป็น
1.73205                                                    1.73205>
```

ซึ่งจะช่วยให้เราได้ทำการเรียกใช้ฟังก์ชั่นต่างๆ ได้จาก Command line ได้ทันที

นอกจากนี้ถ้าเราต้องการจะทราบถึงค่าต่างๆของตัวแปร ที่ถูกใช้ให้เก็บค่าที่ต้องการไว้ในตัวมันเอง ให้มีการแสดงออกมาให้เราได้ทราบ ซึ่งจะสามารถกระทำได้โดยใส่เครื่องหมาย ! นำหน้าตัวแปรที่เราต้องการทราบ เช่น

```
Command : (setq XF (SQRT 3))          <กำหนดค่าให้ XF>
        1.73205                       <รับค่าเข้าไป>
        Command : !XF                 <ให้แสดงค่าที่ถูกเก็บไว้ออกมา>
        1.73205                       <ค่าที่เก็บไว้ในตัวแปร XF>
```

จากหลักการทั้งสองวิธีนี้ เราจะใช้ช่วยในการแก้ไขข้อผิดพลาด (Debug) ในขณะทำการพัฒนาโปรแกรม AutoLISP ซึ่งในการพัฒนาโปรแกรมขึ้นมานั้น ย่อมจะต้องเกิดข้อผิดพลาดขึ้นเสมออย่างหลีกเลี่ยงไม่ได้ และในบางครั้งก็เกิดความผิดพลาดที่หาสาเหตุได้ไม่ชัดเจน ดังนั้นวิธีการหนึ่งที่ช่วยได้ก็คือ เมื่อเกิด error ขึ้นในขณะทำการรัน เราจะใช้วิธีการ Direct Mode ในการเช็คนิพจน์ที่เราสงสัยว่าเป็นต้นเหตุของการเกิด error จากทาง command line หรือเราอาจจะใช้วิธีการใช้เครื่องหมาย ! ในการตรวจสอบค่าของตัวแปรที่ถูกเก็บไว้ ว่าเป็นไปตามค่าที่มันควรจะเป็นหรือไม่ นับได้ว่าวิธีการทั้งสองนี้ช่วยการแก้ไขความผิดพลาดได้ดีทีเดียว

ในการเรียกใช้งานบนฟังก์ชั่นต่างๆของ AutoLISP เราจะต้องสังเกตและจดจำรูปแบบการใช้งานให้แม่น เพราะฟังก์ชั่นแต่ละกลุ่มจะมีรูปแบบที่คล้ายๆกัน เช่น ต้องการเฉพาะตัวเลข ไม่ส่งค่าออกมา ต้องการระยะทางระหว่างจุดสองจุด ฯลฯ ยกตัวอย่าง เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *GETINT* <รับค่าเฉพาะ Integer>

```
Command : (GETINT "Input Numeric : ")          <Enter>
Input Numeric : X                              <ป้อนค่าตัวอักษร>
Requires an integer value                      <แสดงความต้องการตัวเลข>
Input Numeric : 6                              <ป้อนค่าตัวเลข>
6                                              <รับค่า 6 เข้าไป>
```

- *TEXTSCR, GRAPHSCR* <สลับการแสดงโหมด>

```
Command : (TEXTSCR)                            <สวอปไปเท็กซ์โหมด>
Command : (GRAPHSCR)                           <สวอปกลับมา>
```

- *GETDIST* <ต้องการค่าระยะทาง>

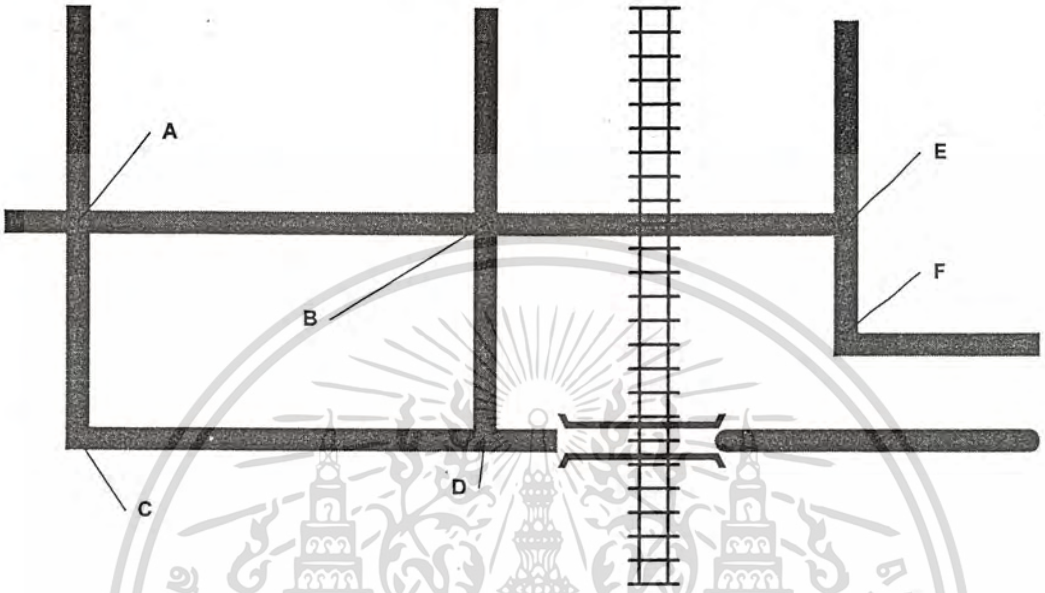
```
Command : (GETDIST "Pict two point : ")
Pick Two Point :                               <pick ตำแหน่งแรกบนสกรีน>
Second Point :                                 <pick ตำแหน่งที่สองบนสกรีน>
7.4561                                         <ระยะที่ได้>
```

2.4 การนำโปรแกรมค้นหาเส้นทางมาประยุกต์เข้ากับภาพแผนที่

แนวคิดของวิทยานิพนธ์นี้เป็นการนำความรู้ความสามารถทางด้านปัญญาประดิษฐ์เข้าประยุกต์เข้ากับภาพแผนที่การจราจรกรุงเทพมหานคร โดยเราจะทำการสมมติว่าสี่แยกต่างๆ ที่อยู่ในกรุงเทพมหานครเป็นจุดโนด ที่เราจะใช้เป็นจุดอ้างอิง ดังนั้นเราจะได้แผนภาพโครงสร้างทางปัญญาประดิษฐ์ที่เอา อัลกอริทึมที่มีอยู่ในปัจจุบันมาประยุกต์ได้

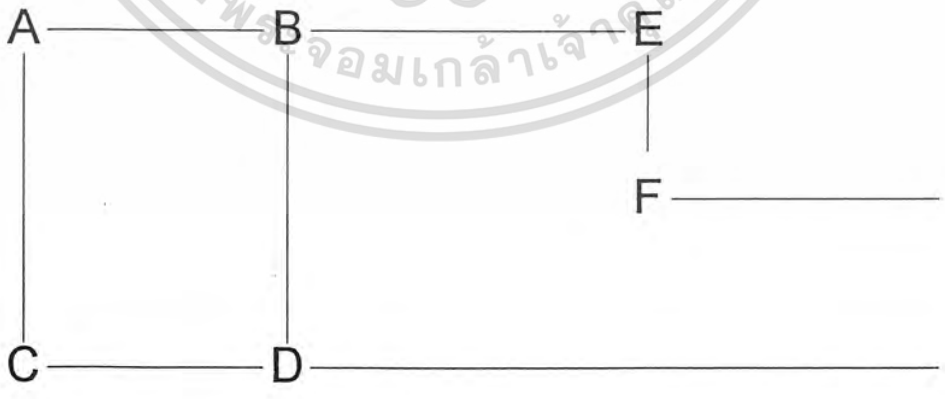
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 อธิบายแนวคิด



รูปที่ 2.4 แสดงถนนที่ใช้เป็นตัวอย่างแนวคิดของวิทยานิพนธ์

จากรูปที่ 2.4 เราจะเห็นว่ามีโนด A B C D E F เป็นตัวแทนในแต่ละแยก ซึ่งเป็นคล้ายกับโนดในแผนภาพข่ายงาน (Network) ได้ดังนี้



รูปที่ 2.5 แสดงถึงการแปลงจากแผนภาพจราจรเป็นแผนภาพข่ายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.5 ทำให้เราสามารถนำทฤษฎีทางด้าน โครงสร้างข้อมูลและปัญญาประดิษฐ์นำมาประยุกต์ได้อย่างมีประสิทธิภาพ

ลักษณะของการทำงานจะเป็นลักษณะของการนำเอาโปรแกรมทางด้านปัญญาประดิษฐ์ที่เขียนขึ้นนำมาหาผลลัพธ์ออกมาในรูปของค่าโคออร์ดิเนตที่เก็บในรูปของเท็กซ์ไฟล์แล้วก็นำมาเข้าโปรแกรมทาง AutoCAD ที่มีแผนภาพจราจรต้นแบบที่สมบูรณ์แล้ว หลังจากนั้นเราก็นำเอาค่าที่หาจากโปรแกรมดังกล่าวแล้ว มาเป็นค่าอินพุต แล้วก็ทำการลากเส้นจากแต่ละจุดโคออร์ดิเนตเพื่อนำมาเป็นค่ามาใช้กับคำสั่งวาดเส้นของ AutoCAD (คำสั่ง line) แล้วก็จะได้เป็นเส้นทางที่ควรเดินทางออกมาอยู่บนภาพจราจรต้นแบบนั้น

ลักษณะการทำงานของโปรแกรมจะต้องทำการค้นหาจุดโนดต่างๆ ได้ แล้วนำค่านี้ไปหาว่าเป็นค่าโคออร์ดิเนตเท่าไหร่ หลังจากนั้นก็นำเข้าไปเก็บเป็นไฟล์เพื่อรอการเรียกต่อไปได้

ส่วนค่าโคออร์ดิเนตเราจะได้จากการทำการ Digitize ภาพแผนที่ให้เรียบร้อยแล้วก็มาหาค่าตรงกลางจุดแยกต่างๆ เช่น สี่แยก สามแยก หัวมุมเลี้ยว เป็นต้น เพื่อนำค่านั้นสมมติให้เป็นตัวอักษรชุดหนึ่งแล้วนำไปเป็นจุดโนดเพื่อให้โปรแกรมสามารถทำงานได้อย่างไม่มีปัญหา และเมื่อได้ผลลัพธ์เป็นโนดที่เรียงกันเป็นเส้นทางต่างๆแล้วก็นำมาเทียบตำแหน่งกับข้อมูลเดิมเพื่อหาว่าค่าของโนดนั้นเป็นตำแหน่งโคออร์ดิเนตที่เท่าไหร่ค่อยส่งไปให้ AutoCAD ทำการลากเส้นเชื่อมต่อจุดแต่ละจุด ผลลัพธ์ก็จะได้เป็นเส้นทางที่ต้องการนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

ขั้นตอนในการดำเนินงาน

ขั้นตอนการดำเนินงานแบ่งออกเป็น ขั้นตอนดังต่อไปนี้

1. กำหนดขอบเขตของการศึกษา และจุดประสงค์ต่างๆในวิทยานิพนธ์นี้
2. เก็บรวบรวมข้อมูลต่างๆที่จำเป็นต้องใช้ในการวิเคราะห์ ซึ่งเป็นการออกไปรวบรวมข้อมูลเองและเป็นข้อมูลทางทุติยภูมิที่ได้มีการสำรวจและรวบรวมไว้แล้ว โดยหน่วยงานราชการและเอกชน และนำมาเก็บเป็นฐานข้อมูลที่ใช้ในอนาคตต่อไป
3. ค้นคว้าเพิ่มเติมเกี่ยวกับหลักการในการเขียนโปรแกรมที่ใช้ในการค้นหาเส้นทางต่างๆ ตลอดจนโปรแกรมและอุปกรณ์ที่ใช้ในการเก็บภาพแผนที่ว่าควรจะใช้โปรแกรมอะไร อุปกรณ์อะไร เพราะอะไร และหาความแตกต่างของการใช้วิธีในการค้นหาเส้นทางที่แตกต่างกันแล้วสามารถสรุปได้ว่าวิธีที่ใช้มาทั้งหมดวิธีไหนเหมาะกับงานมากกว่ากัน
4. ทำการเก็บภาพแผนที่เข้าเครื่องคอมพิวเตอร์ โดยใช้แผนที่มาตราส่วน 1 : 1000
5. เขียนโปรแกรมค้นหาเส้นทางด้วยภาษา Prolog เพื่อใช้ในการเปรียบเทียบการทำงานที่เหมาะสมที่สุดกับการศึกษานี้ โดยได้มีการกำหนดจุดต้นและจุดปลายของเส้นทาง เพื่อใช้ในการหาเส้นทางที่ใช้เวลาน้อยที่สุดในการเดินทาง
6. ทดสอบความถูกต้องของโปรแกรมที่เขียนขึ้นโดยการจำลองสถานการณ์ต่างๆที่จะเกิดขึ้นในการเดินทางได้ ตลอดจนหาความแตกต่างของแต่ละโปรแกรมว่าวิธีใดเหมาะสมที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 การรวบรวมข้อมูล

ข้อมูลต่างๆที่นำเข้ามาใช้นี้เป็นข้อมูลที่ต้องออกไปรวบรวมเองเป็นส่วนใหญ่เนื่องจากว่าไม่มีองค์การหรือหน่วยราชการใดๆที่ทำหน้าที่ในการรวบรวมข้อมูลเกี่ยวกับเส้นทางการจราจรอย่างแท้จริง ดังนั้นการทำงานจึงเป็นไปอย่างไม่ค่อยสะดวกและข้อมูลที่ได้จึงไม่สามารถรับประกันได้ว่าถูกต้อง แต่ถึงกระนั้นข้อมูลบางอย่างที่ได้ก็เป็นข้อมูลที่ได้จากหน่วยราชการ ดังนั้นจึงพอจะเชื่อถือได้ว่าเป็นข้อมูลที่ดีที่สุดที่พอจะหาได้แล้ว เราจะแบ่งประเภทของข้อมูลที่ได้จากทางราชการหรือหน่วยงานองค์กรใดๆ ได้ดังต่อไปนี้คือ

ข้อมูลที่ได้จากหน่วยงานราชการ คือ ภาพแผนที่ถนนเยาวราช ซึ่งให้มาทั้งหมด 4 แผ่น เป็นแผนที่ที่ครอบคลุมเนื้อที่ประมาณ 3,330 ตารางกิโลเมตร โดยจะเน้นว่าให้ถนนเยาวราชเป็นศูนย์กลางในการศึกษาเส้นทางการเดินทาง

ข้อมูลที่ได้ทำการรวบรวมเอง คือ

1. เส้นทางการเดินรถ ซึ่งต้องออกไปทำการสำรวจเองทำให้ได้ข้อมูลในถนนแต่ละเส้นทางว่า
   - ถนนเยาวราช เป็นถนนที่ทำการเดินรถในเลนเดียวเท่านั้น
   - ถนนอื่นๆที่อยู่รอบๆ ส่วนใหญ่เป็นถนนที่มีเส้นทางการเดินรถแบบทั้งเลนเดียวและสองเลนแต่ส่วนมากมักเป็นการวิ่งเลนเดียว
   - ตรอก ซอยจำนวนมาก ถ้าดูตามแผนที่จะเห็นว่าเป็นซอย แต่จากการไปสำรวจดูตามสภาพความเป็นจริงแล้ว จะพบว่าส่วนมากเป็นซอยที่มีสิ่งของวางระเกะระกะเป็นจำนวนมาก ทำให้รถวิ่งไม่ได้

2. สภาพถนน ซอย ตรอก
   - สภาพถนนเยาวราช มักมีสภาพถนนที่ค่อนข้างดี คือไม่ค่อยจะต้องมีการซ่อมแซมบ่อยนัก แต่ก็มีบ้างในกรณี เช่น วางสายโทรศัพท์ วางสายไฟใหม่ เป็นต้น จึงสมมติให้สภาพถนนเยาวราชเป็นสภาพถนนที่ไม่มีการเปลี่ยนแปลง
   - ถนนอื่นๆที่อยู่รอบๆ มักเป็นถนนที่มีสภาพที่ดี แต่ก็มีอยู่บ้างที่มีสภาพค่อนข้างแย่ คือ มีการซ่อมแซมที่บ่อยครั้ง และมีการปรับปรุงเปลี่ยนแปลงเส้นทางการเดินรถจราจรบ่อยๆ ทำให้ข้อมูลที่ได้อาจไม่ตรงตามความเป็นจริงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตรอก ซอย ที่อยู่บริเวณถนนเยาวราชมักมีสิ่งของวางระเกะระกะเป็นจำนวนมาก ทำให้ไม่สามารถให้รถมอเตอร์ไซด์วิ่งได้ จึงต้องมีการสำรวจเอง และประกอบกับการทำงานในการสำรวจเอง เป็นไปได้อย่างลำบากเพราะว่าไม่สามารถเดินเข้าไปสำรวจ บริเวณซอย ตรอก ที่อยู่ลึกๆ และมีของวางระเกะระกะตลอดทาง ข้อมูลที่ได้ในส่วนนี้จึงอาจมีข้อผิดพลาดได้

3. อัตราเร็วของรถที่วิ่ง

- บริเวณถนนเยาวราช เรากำหนดอัตราความเร็วให้ที่ 30 กิโลเมตรต่อชั่วโมงตลอดสาย ข้อมูลนี้ได้มาจากการบันทึกช่วงเวลาที่ใช้ในการออกไปสำรวจเดินทางและระยะทางที่คงที่ตลอดเป็นจำนวน 25 ครั้ง นำมาหาความเร็วของรถที่วิ่งในแต่ละครั้ง แล้วนำมาหาค่าเฉลี่ยความเร็วรถจากทั้งหมดจะได้ค่าประมาณ 29.2356 กิโลเมตรต่อชั่วโมงจึงปัดให้เป็น 30 กิโลเมตรต่อชั่วโมง แต่ทว่าการออกไปสำรวจแต่ละครั้งเป็นการออกไปสำรวจในช่วงเวลาที่แตกต่างกัน วันต่างกัน ทำให้ข้อมูลที่ได้เป็นการประมาณอย่างคร่าวๆเท่านั้นไม่ใช่ค่าจริงๆที่เกิดขึ้นในช่วงเวลานั้น ในวันนั้นๆ
- อัตราเร็วของรถในแต่ละซอยเป็นอัตราที่ไม่แน่นอน แต่เราจะทำการสมมติจากสถิติและการซุ่มชื้นมาสำรวจว่ามีอัตราเฉลี่ยของรถเป็นเท่าไหร่ จึงเป็นค่าที่อาจผิดพลาดได้ แต่เป็นวิธีการที่พอจะคาดหวังได้ว่าใกล้เคียงจากความเป็นจริงได้
- อัตราเร็วของรถในถนนแต่ละสายรอบๆ ถนนเยาวราช ก็จะได้จากสูตรการหาค่าน้ำหนัก (Weight) จากการคำนวณของสูตร

  $T = (S/V) * W$

  เมื่อ T คือเวลาที่ใช้ในการวิ่งซึ่งก็คือค่าที่จะนำมาพิจารณา

  S คือระยะทางระหว่างจุด 2 จุดใดๆ

  V คือความเร็วของรถมอเตอร์ไซด์ (มีค่า 30 กิโลเมตรต่อชั่วโมง)

  W มีค่าเป็น

  1 เมื่อเป็นเส้นทางที่มีสภาพการจราจรคล่องตัว

  1.5 เมื่อเป็นเส้นทางที่มีสภาพการจราจรติดขัดบ้างบางครั้ง

  2 เมื่อเป็นเส้นทางที่มีสภาพการจราจรติดขัดเกือบตลอดเวลา

  ก็จะได้ค่าน้ำหนัก (Weight) ของแต่ละเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ซอยตัน และ ตรอกตัน

- เนื่องจากว่าซอยบางซอยเป็นซอยที่ปรากฏทางเข้าหนึ่งในแผนที่แต่ปรากฏว่าไม่สามารถหาทางออกให้กับเส้นทางดังกล่าวได้เนื่องจากว่าเส้นทางดังกล่าวเป็น ซอยตันบ้าง เป็นซอยที่เข้าไปเป็นที่อยู่อาศัยบ้าง เป็นสถาที่ราชการ เป็นสถานีบริการน้ำมัน เป็นซอยที่ตัดไปชนกับคลอง เป็นซอยที่เข้าไปมีซอยแยกมากมาย แตกย่อยออกไปอีก แต่สุดท้ายก็ไม่สามารถหาทางออกไปที่อื่นได้ แม้ว่าจะมีซอยแยกออกไปอีกมากมาย เนื่องจากว่าเมื่อดูไปตามเส้นทางไปเรื่อยๆ ก็จะพบว่ากลับมาวนที่เดิมอีกจนได้ เป็นต้น เราจึงสมมติให้ซอยนั้นเป็นทางตัน ตั้งแต่ทางเข้าและจะไม่คิดเป็นเส้นทางในการเดินทางได้เลย

3.2 การติดตั้ง AutoLISP

3.2.1 การติดตั้งพื้นที่หน่วยความจำใน AutoLISP

โดยปกติ AutoLISP จะจัดการหน่วยความจำทั้งสองส่วนด้วยตัวมันเอง ซึ่งทั้งสองส่วนนั้นคือ Stack และ Heap โดยพื้นที่ในสแต็ก คือ ที่ที่ AutoLISP เก็บอาร์กิวเมนต์ของคำสั่ง (Function Argument) และการคำนวณเฉพาะหน้า ส่วน Heap จะทำหน้าที่เก็บคำสั่งและตัวแปรทั้งหมด เมื่อเราใช้ AutoCAD Release 12 เราจะมีค่าที่ตั้งเอาไว้อยู่แล้วว่า จะตั้งค่า Heap ไว้ที่ 40,000 ไบต์ และตั้งค่า Stack ไว้ที่ 3,000 ไบต์ แต่มีข้อจำกัดว่าขนาดรวมของหน่วยความจำทั้งสองต้องมีพื้นที่รวมกันไม่เกิน 45,000 ไบต์ และถ้าในขณะที่เรากำลังใช้ AutoLISP แล้วเราใช้พื้นที่เกินกว่านี้ คำสั่งต่างๆจะหยุดทำงานและจะมีข้อความว่า พื้นที่โหนดไม่เพียงพอ (Insufficient node space)

วิธีการแก้ไขทำได้โดยการเพิ่มบรรทัดต่อไปนี้ลงในไฟล์ AUTOEXEC.BAT โดยเริ่มจากการตั้งค่า LISPHEAP ไว้ที่ 40,000 ไบต์ และค่า LISPSTACK ไว้ที่ 5,000 ไบต์ เพื่อเป็นค่าเริ่มต้นเมื่อเริ่มเราเรียกโปรแกรมจากใน DOS

```
SET LISPHEAP=40000
SET LISPSTACK=5000
SET ACADFREERAM=24
```

(โดยที่ต้องระวังว่าข้างหน้าหรือข้างหลังเครื่องหมายเท่ากับ (=) ห้ามมีช่องว่าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่ตั้งให้ ACADFREERAM ในบรรทัดที่8 จะจองเนื้อที่หน่วยความจำไว้ 24 กิโลไบต์ สำหรับเป็นพื้นที่ที่เก็บไว้ใช้ในขณะทำงาน โดยที่คำสั่งเหล่านี้จะมีผลกับพื้นที่ที่จองไว้สำหรับ AutoCAD เท่านั้น โดยเราจะไม่เสียหน่วยความจำใดๆ ขณะที่ทำงานโปรแกรมอื่นๆ

3.2.2 การตั้งค่า BUFFER และ FILE ใน DOS

ไฟล์ CONFIG.SYS ใช้ในการระบุลักษณะในการติดต่กับ DOS โดยเราสามารถเพิ่มความเร็วในการทำงานของระบบของเราได้มากกว่า 20% ถ้าเราเพิ่มพื้นที่ในการทำงานในหน่วยความจำเพื่อให้ DOS สามารถใช้ในการอ่านและเขียนในฮาร์ดดิสก์ได้

เราเพิ่มข้อความในไฟล์ CONFIG.SYS โดยใช้ editor ตัวใดก็ได้แล้วเก็บไว้เป็นไฟล์ข้อความ (text file) โดยข้อความที่เพิ่มมีดังนี้

```
FILES=22
BUFFERS=32
```

คำสั่ง FILES ทำหน้าที่บอก DOS ว่าจะมีอยู่กี่ไฟล์ที่สามารถใช้ได้ในเวลาเดียวกัน ถ้าไม่ได้ถูกกำหนดไว้ ก็จะถูกตั้งไว้ให้ค่าเป็น 8

คำสั่ง BUFFERS เป็นการระบุจำนวนของพื้นที่ที่ใช้ในการทำงานในหน่วยความจำที่ DOS เปิดเพื่อใช้ในการอ่านจากหรือเขียนลงในดิสก์ ถ้าไม่ได้ระบุเอาไว้ก็จะถูก DOS ตั้งไว้ให้มีค่าเป็น 2

การตั้งให้มีค่าไฟล์เป็น 22 และ ค่าบัฟฟเฟอร์เป็น 32 ก็เพื่อพัฒนาประสิทธิภาพของฮาร์ดดิสก์ให้สูงขึ้น แต่เราอาจจะตั้งให้มีค่าอื่นก็ได้ถ้าได้ผลที่ดีกว่า

3.2.3 การติดตั้ง editor ให้ AutoCAD

โดยปกติแล้วโปรแกรม AutoCAD จะติดตั้งโปรแกรม editor ให้ผู้ใช้สามารถเรียกใช้ได้ทันทีอยู่แล้วภายใต้คำสั่งของ Command Line โดยการเรียก

Command Line : edit	<แล้วกด Space bar หรือ Enter>
File to edit : (file name)	<พิมพ์ชื่อไฟล์พร้อม path>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากเราพิมพ์คำสั่ง edit แล้วกด Space Bar เครื่องก็จะขึ้นบรรทัดใหม่ เพื่อถามว่าต้องการแก้ไขไฟล์ใด เพื่อให้เราป้อนชื่อไฟล์ แต่ถ้าเรากด Enter โปรแกรมก็จะขึ้นบรรทัดใหม่ 2 บรรทัดแต่ก็จะถามเหมือนเดิม

เรามีทางเลือกที่จะใช้โปรแกรม editor ตัวอื่นมาใช้ในการทำงาน อย่างเช่นในที่นี้ขอแนะนำให้ใช้ text editor ตัวหนึ่งที่ชื่อว่า Q-editor โดยที่ต้องหาตัวโปรแกรมมาเอง โดยทั่วไปก็คือ Q.EXE เนื่องจากเป็นโปรแกรมที่มีขนาดไม่ใหญ่มากนัก และการใช้งานก็ง่ายต่อการใช้งาน วิธีการติดตั้งก็ทำโดยการติดตั้งลงในไฟล์ที่ชื่อว่า ACAD.PGP

ถ้าเราลองพิมพ์คำสั่งในไฟล์ ACAD.PGP ออกมาจะพบว่า ไฟล์นี้เป็น test file ทำให้เราสามารถแก้ไขได้ทันที เมื่อเราสั่งพิมพ์ออกมาดูโดยใช้คำสั่งต่อไปนี้

```
C:\> CD\ACAD
C:ACAD>TYPE ACAD.PGP
```

เราก็จะได้รายการต่างๆดังต่อไปนี้

```
CATALOG,DIR/W,30000,*Files : ,0
DEL,DEL,30000,File to delete : ,0
DIR,DIR,30000,File specification : ,0
EDIT,EDLIN,42000,File to edit : ,0
SH, ,30000,*DOS Command : ,0
SHELL, ,12700,*DOS Command : ,0
TYPE,TYPE,30000,File to list : ,0
```

สังเกตว่ามีการแสดงรายชื่อว่ามี EDLIN เข้าไปอยู่ด้วย แต่ถ้าเครื่องที่เรามียังไม่มีบรรทัดที่มี EDLIN นี้เราก็มีความจำเป็นที่จะต้องพิมพ์เพิ่มเข้าไป แต่ถ้ามีอยู่แล้วและลักษณะก็คล้ายๆกัน ก็ไม่ต้องพิมพ์เพิ่มเข้าไป แต่อย่างไรก็ตามเราสามารถทำการแก้ไขในส่วนนี้ได้ถ้าเราไม่ถนัดที่จะใช้ EDLIN เราก็สามารถแก้ไขให้เป็นอย่างอื่นได้แต่มีเงื่อนไขที่ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรมที่เพิ่มเข้าไปในไฟล์ ACAD.PGP ต้องมีขนาดไม่เกิน 50 กิโลไบต์
2. ต้องทำการกำหนด PATH ของโปรแกรมด้วยว่าอยู่ subdirectory อะไร (ถ้ากำหนด PATH ใน AUTOEXEC.BAT ไว้แล้วว่ามี PATH นั้นอยู่ก็ไม่ต้องกำหนดก็ได้ แต่เพื่อความแน่นอนก็ควรกำหนดไว้)

อย่างเช่นในกรณีที่ต้องการใช้โปรแกรม Q.EXE เราก็ต้องดูขนาดของไฟล์ก่อนว่ามีขนาดเท่าใด และทำการพิมพ์เข้าไป เช่น

```
EDIT,C:>\Q.EXE,52892,File to edit : ,0
```

คราวนี้เมื่อเราพิมพ์คำสั่ง EDIT เข้าไป เครื่องก็จะปรากฏข้อความว่า File to edit : เพื่อให้เราป้อน File name (ชื่อไฟล์) และเมื่อเราป้อนชื่อไฟล์เสร็จ ไฟล์นั้นก็จะเข้าไปที่โปรแกรม test editor ที่เราได้กำหนดไว้ทันที

3.2.4 การเขียนโปรแกรมเชื่อมต่อการทำงาน

การเขียนโปรแกรม AutoLISP นี้เพื่อทำการอ่านค่าโคออร์ดิเนตจากไฟล์ข้อความ (text file) แล้วทำการลากเส้นเพื่อเชื่อมต่อเส้นทางที่เกิดจากโคออร์ดิเนตที่ได้จากการทำงานในส่วนโปรแกรม Prolog ซึ่งเป็นส่วนการค้นหาภาพแผนที่

ขั้นตอนการเขียนโปรแกรมมีดังต่อไปนี้

1. อ่านไฟล์ที่เป็นไฟล์ข้อความแล้วเก็บค่าที่อ่านได้ไว้ตัวแปรตัวหนึ่งโดยการใช้คำสั่ง

```
(setq ifp (open in "r")
```

2. ทำการตั้งค่าว่าเส้นใดๆ ต่อไปนี้ เป็นเส้นที่มีสีเหลือง โดยค่าของสีเหลืองคือค่าหมายเลข 2 ของคำสั่ง color โดยมีรูปแบบคือ

```
(command "color" "2")
```

โดยคำสั่ง command คือคำสั่งที่ใช้ในการเรียกคำสั่งที่อยู่ใน AutoCAD อยู่แล้วมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. อ่านค่าที่ได้ออกมาเก็บในตัวแปรสองตัวสมมติให้เป็น x และ y แล้วนำค่านั้นเป็นค่าโคออร์เนตเพื่อใช้ในการวาดเส้นทางให้มีสีตามที่ต้องการ โดยการใช้คำสั่ง

(setq x (read-line ifp)) เป็นการอ่านค่ามาเก็บในตัวแปร x

(setq y (read-line ifp)) เป็นการอ่านค่ามาเก็บในตัวแปร y

แต่การลากเส้นต้องทำการลากจาก x ไป y แล้วก็นำค่า y นั้นมาเป็นค่า x เพื่อจะได้ลากเส้นจาก x ไป y อีก ไปเรื่อยๆจนกว่าจะหมดไฟล์ ข้อสังเกตที่สำคัญคือ รูปแบบของไฟล์ข้อความนั้น ค่าโคออร์ดิเนตต้องเก็บแยกบรรทัดกัน คือ บรรทัดละค่าเท่านั้น โดยค่าที่เก็บมีรูปแบบต่อไปนี้คือ เป็นในรูปของแกน x และแกน y คั่นกลางด้วยเครื่องหมาย comma (,) เช่น 3,5 8,5 เป็นต้น

4. ดังนั้นเราจะได้โปรแกรมเต็มๆดังนี้

```
(defun data (in)
    (setq ifp (open in "r"))
    (command "color" "2")
    (setq x (read-line ifp))
    (while (setq y (read-line ifp))
        (command "line" x y "")
        (setq x y)
    )
    (close ifp)
)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเขียนโปรแกรมภาษา Prolog

3.3.1 การเขียนโปรแกรมการค้นหาแบบตามแนวลึกก่อน (Depth-First Search)

การค้นหาแบบตามแนวลึกก่อนเป็นการค้นหาแบบไม่มี Heuristic ทำให้การทำงานของโปรแกรมนี้ค่อนข้างจะเป็นแบบพื้นฐาน แต่เราจะต้องระวังให้มากในเรื่องต่างๆ ดังต่อไปนี้ คือ

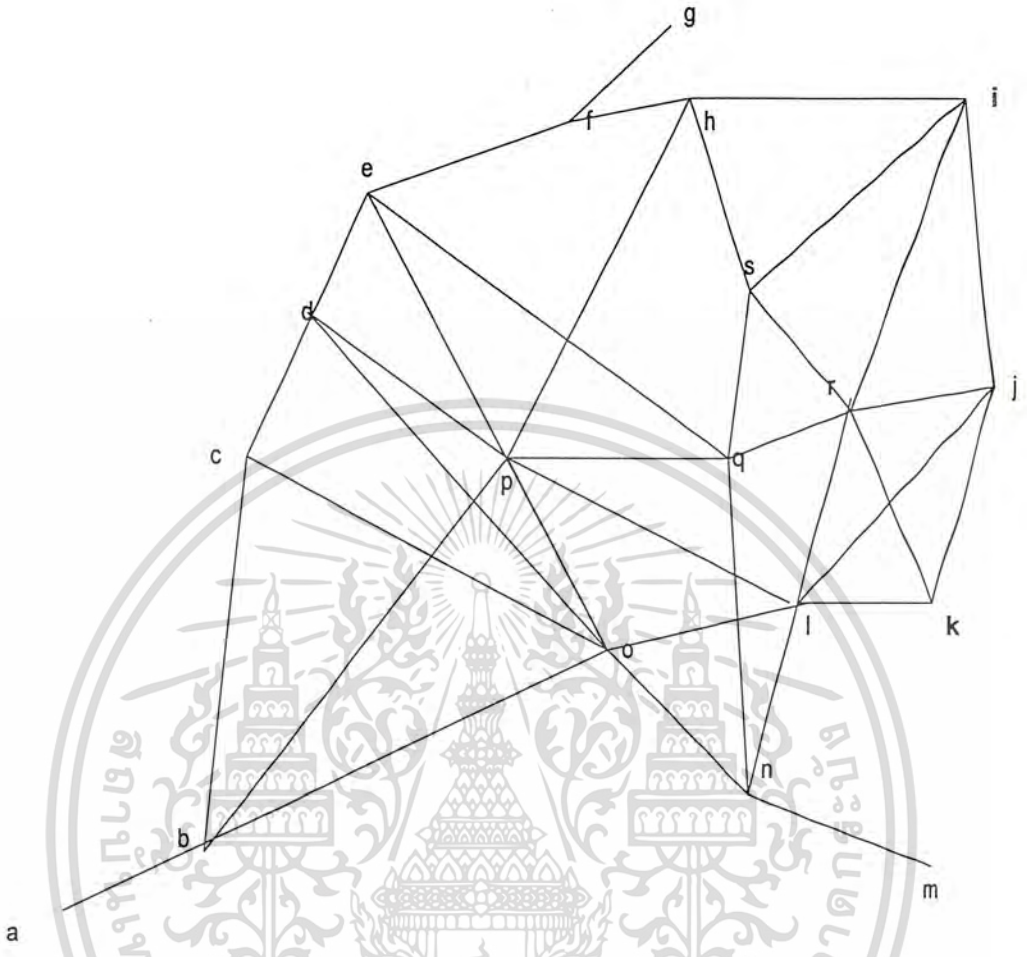
1. ระวังเรื่องการวนหาแบบไม่มีที่สิ้นสุด (Circular Loop)
2. ระวังเรื่องการจัดลำดับฐานข้อมูลภายใน (Internal DataBase)

ดังนั้นเพื่อทำการแก้ไขแล้ว จำเป็นต้องมีการตรวจสอบการทำงานหลังจากเขียนโปรแกรมเสร็จเรียบร้อยแล้ว แต่ทว่าการแก้ไขในส่วนนี้เป็นส่วนที่เครื่องไม่สามารถตรวจสอบข้อผิดพลาดแบบนี้ได้ ดังนั้นจึงต้องมีการตรวจสอบโดยตัวผู้เขียนโปรแกรมเอง

ตัวอย่างการเขียนโปรแกรมนี้เราเริ่มต้นจากสมมติเส้นทางเพื่อใช้ในการเขียนโปรแกรมการค้นหาเส้นทางแบบตามแนวลึกก่อน ดังต่อไปนี้

สมมติให้เป็นการศึกษาเส้นทางการเดินทางจากโหนดต่างๆดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงตัวอย่างเส้นทางการเดินทางแบบที่ 1

ซึ่งสามารถอธิบายได้ดังต่อไปนี้

1. กำหนดโครงสร้างฐานข้อมูลของฐานความรู้ที่ต้องใช้มีอยู่ 2 ตัว ดังนี้

- โครงสร้างการเดินทาง โดยให้เป็นการเดินทางแบบไปและกลับได้ หมายถึง สามารถเดินทางจาก a ไป b ได้ และสามารถเดินทางจาก b ไป a ได้เช่นกัน ดังนั้นถ้าเราสมมติว่าตัวที่ทำหน้าที่พิจารณาเส้นทางการเดินทางเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

move_to(symbol, symbol, integer)

ตัวอย่างเช่น move_to(e, f, 338)
หมายถึง การเดินทางจากจุด e ไปจุด f โดยมีระยะทางเป็น 338 หน่วย เป็นต้น

- กำหนดโครงสร้างตัวที่สองเรียกว่า visited ทำหน้าที่เก็บเส้นทางการเดินทางของเมืองที่ผ่าน ซึ่งโครงสร้างตัวนี้จะให้คุณเห็นเส้นทางที่เกิดจากกลไกการทำงานภายใน โดยแสดงให้เห็นว่ามีเมืองอะไรบ้างที่ได้รับการตรวจสอบก่อนที่จะเป็นผลลัพธ์แต่ละตัว และยังแสดงให้เห็นเส้นทางรวมที่เกิดขึ้นทั้งหมดด้วย ดังนั้นบางที visited จะทำหน้าที่เก็บเมืองที่ไม่อยู่บนเส้นทางการเดินทางสุดท้ายด้วย เพราะว่าผลลัพธ์ที่ได้อาจจะเป็นเส้นทางตัน (dead ends) ก็คือได้มีการตรวจสอบเมืองต่างๆแล้วแต่ไม่สามารถไปหาจุดสุดท้ายที่ต้องการได้ สิ่งนี้จะแสดงให้เราเห็นว่าแต่ละเทคนิคมีประสิทธิภาพในการเชื่อมต่อความสัมพันธ์ระหว่างสิ่งต่างๆในปัญหาได้อย่างไร และเราสามารถกำหนดโครงสร้างได้เป็น

visited(symbol)

ดังนั้นเราจะได้ส่วนแรกของโปรแกรมเป็น

```
database
        move_to(symbol, symbol, integer)
        visited(symbol)
```

2. กำหนดสิ่งที่ทำหน้าที่หาเส้นทางระหว่างเมืองสองเมือง เช่น ระหว่างเมือง a และเมือง g เป็นต้น ซึ่งเราเรียกสิ่งนี้ว่า data โดยมีการทำงานดังต่อไปนี้

```
data(T, T2, D):-
        move_to(T, T2, D),
        add_to_route(T),
        retractall(move_to(T, T2, D)).
data(T, T2, D):-
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        move_to(T, X, D2),
        X <> T2,
        add_to_route(T),
        retractall(move_to(T, X, D2)),
        data(X, T2, D3),
        D = D2+D3.
data(T, _, D):-
        write("Dead end at ", T),
        nl,
        D = 0,
        fail.
```

โดยเราสามารถอธิบายการทำงานของโปรแกรมได้ดังต่อไปนี้

1. เงื่อนไขแรกจะทำงานได้ถ้ามีเส้นทางการเดินทางระหว่างเมือง T และ T2 และถ้ามีเส้นทางการเดินทางระหว่างเมืองทั้งสองอยู่จริงแล้ว ก็จะรับค่าระยะห่างระหว่างเมืองเป็น D โดยที่ระยะห่างนี้เราต้องกำหนดให้อยู่ในระบบฐานข้อมู ลของเราอยู่แล้ว หลังจากนั้นก็เอาค่า T นั้น ไปเก็บไว้ที่ระบบฐานข้อมูล visited ดังนั้นถ้าเราเรียกใช้เงื่อนไขนี้ด้วย

```
data(h, p, X)
```

แล้วเงื่อนไขแรกก็จะเป็นจริง และเราก็จะได้ค่า X ออกมาว่าเป็น 628
อย่างไรก็ตามลองนึกภาพว่า ถ้าเราเรียกเงื่อนไขนี้ด้วย

```
data(a, p, X)
```

ถ้าเป็นในกรณีนี้แล้ว เงื่อนไขแรกจะได้ค่าออกมาเป็นเท็จเพราะว่าไม่มีเส้นทางที่เชื่อมต่อระหว่างเมืองทั้งสองเมืองนี้ ดังนั้นก็จะไปพิจารณาในเงื่อนไขที่สองแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เงื่อนไขที่สองจะพยายามหาเส้นทางการเชื่อมต่อระหว่างเมืองที่เป็นจุดเริ่มต้น และเมืองใดๆก็ตาม ซึ่งถ้าพบว่ามีการเชื่อมต่อได้แล้ว ก็จะเรียกให้มีการทำงานแบบเรียกตัวมันเอง คือจะเรียกให้ **data** ทำงานอีกครั้งหนึ่ง เพื่อดูว่าเมืองอื่นๆ ที่เชื่อมต่อจากเมืองที่เป็นจุดเริ่มต้นนั้น สามารถเชื่อมต่อไปยังเมืองที่เป็นจุดหมายปลายทางได้หรือไม่ ซึ่งถ้าสามารถเชื่อมต่อได้ ก็จะทำให้การเรียกตัวมันเองที่ไปเรียกเงื่อนไขแรกเป็นจริง แล้วกลับมาทำในเงื่อนไขที่สองในส่วน ที่เหลือต่อไป เพื่อเอาจำนวนระยะห่างที่ได้มารวมกัน สังเกตว่าเงื่อนไขที่สองจะรวมเมืองต่างๆเก็บไว้ในฐานข้อมูล **visited** ดังนั้นโปรแกรมจะหาว่ามีเส้นทางใดจากเมือง a ไปเมือง p หรือไม่ และขอให้ตรวจสอบการทำงานของ **data** ให้ดี เพราะว่านี่คือส่วนที่ทำหน้าที่เป็นการค้นหาแบบตามแนวลึกก่อนของฐานความรู้ สิ่งหนึ่งที่เราต้องสนใจเป็นพิเศษนั่นคือการป้องกันไม่ให้เกิดการวนรอบซ้ำ ซึ่งในที่นี้เราป้องกันโดยการใช้การดึงเอาเส้นทางที่เคยใช้ในการหาเส้นทางออกไปจากระบบภายใน ทำให้การหาเส้นทางครั้งต่อไปจะไม่สามารถหาเส้นทางจากเมือง a ไปยังเมือง l อีกได้เพราะว่าเราเคยหาไปแล้วครั้งหนึ่งและไม่ต้องการให้เกิดการค้นหาอีกเพราะว่าจะเป็นต้นเหตุให้เกิดการวนแบบไม่มีที่สิ้นสุด ซึ่งการเอาข้อมูลในระบบฐานความรู้ออก เราใช้คำสั่ง

```
retractall(move_to(T, X, D2))
```

ซึ่งหมายถึงการเอาข้อมูลที่เป็นการเชื่อมต่อระหว่างเมืองที่อยู่ในตัวแปร T และตัวแปร X ที่มีระยะห่างระหว่างเมืองเป็น D2 ออกไปจากระบบฐานข้อมูลภายใน

3. โปรแกรมจะใช้การระบุจุดมุ่งหมาย (goal) ที่อยู่ภายใน ให้เป็นทั้งตัวกำหนดระบบฐานความรู้เริ่มต้นในการเดินทาง และเริ่มการทำงานของโปรแกรม ดังนั้นเราสามารถระบุจุดมุ่งหมาย ได้เป็น

```
goal
        assert(move_to(e, d, 164)),
        assert(move_to(e, p, 558)),
        assert(move_to(e, q, 526)),
        assert(move_to(e, f, 338)),
        assert(move_to(k, j, 263)),
        assert(move_to(k, l, 403)),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
assert(move_to(k, r, 501)),
assert(move_to(h, p, 628)),
assert(move_to(h, s, 219)),
assert(move_to(h, i, 345)),
assert(move_to(h, f, 182)),
assert(move_to(p, e, 558)),
assert(move_to(p, h, 628)),
assert(move_to(p, l, 489)),
assert(move_to(p, b, 1104)),
assert(move_to(p, d, 497)),
assert(move_to(p, o, 304)),
assert(move_to(p, q, 311)),
assert(move_to(d, p, 497)),
assert(move_to(d, e, 164)),
assert(move_to(d, o, 740)),
assert(move_to(d, c, 225)),
assert(move_to(j, k, 263)),
assert(move_to(j, i, 394)),
assert(move_to(j, r, 155)),
assert(move_to(j, l, 384)),
assert(move_to(g, f, 180)),
assert(move_to(o, p, 304)),
assert(move_to(o, d, 740)),
assert(move_to(o, c, 629)),
assert(move_to(o, l, 361)),
assert(move_to(o, n, 328)),
assert(move_to(o, b, 951)),
assert(move_to(f, e, 338)),
assert(move_to(f, h, 182)),
assert(move_to(f, g, 180)),
assert(move_to(a, b, 339)),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
assert(move_to(b, c, 805)),
assert(move_to(b, o, 951)),
assert(move_to(b, p, 1104)),
assert(move_to(b, a, 339)),
assert(move_to(q, p, 311)),
assert(move_to(q, e, 526)),
assert(move_to(q, s, 185)),
assert(move_to(q, r, 280)),
assert(move_to(q, n, 582)),
assert(move_to(m, n, 134)),
assert(move_to(c, d, 225)),
assert(move_to(c, o, 629)),
assert(move_to(c, b, 805)),
assert(move_to(s, r, 185)),
assert(move_to(s, i, 479)),
assert(move_to(s, h, 219)),
assert(move_to(s, q, 185)),
assert(move_to(n, l, 442)),
assert(move_to(n, q, 582)),
assert(move_to(n, m, 134)),
assert(move_to(n, o, 328)),
assert(move_to(l, r, 440)),
assert(move_to(l, p, 489)),
assert(move_to(l, j, 263)),
assert(move_to(l, k, 403)),
assert(move_to(l, o, 361)),
assert(move_to(l, n, 442)),
assert(move_to(r, i, 464)),
assert(move_to(r, s, 185)),
assert(move_to(r, l, 440)),
assert(move_to(r, j, 155)),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
assert(move_to(r, k, 501)),
assert(move_to(r, q, 280)),
assert(move_to(i, h, 345)),
assert(move_to(i, s, 479)),
assert(move_to(i, r, 464)),
assert(move_to(i, j, 394)),
findroute, nl.
```

4. สังเกตว่าตัวเพรดิเคต findroute เป็นตัวทำหน้าที่ถามเราว่าให้จุดเริ่มต้นเป็นอะไร และจุดสิ้นสุดเป็นอะไร นั่นก็หมายความว่า เราสามารถใช้โปรแกรมในการหาเส้นทางระหว่างเมืองสองเมืองได้

5. แต่การทำงานของโปรแกรมนี้เราต้องการให้สามารถเขียนค่าโคออร์ดิเนตของแต่ละโหนดที่เป็นเส้นทางต่างๆลงในไฟล์ได้ ดังนั้นเราต้องทำการเปิดไฟล์เพื่อทำการเขียนลงไฟล์ โดยสามารถทำได้โดยการใช้คำสั่งต่อไปนี้

```
openwrite(filetype, "ddepth.txt"),
writedevice(filetype),
not(writeroute),
coordinates(B, C),
write(C),
closefile(filetype),
writedevice(screen),
not(displayroute),
write(B).
```

โดยทำการเก็บลงในไฟล์ชื่อ ddepth.txt และในการเขียนลงในไฟล์ได้เราต้องมีการค้นหาว่าโหนดนั้นมีค่าโคออร์ดิเนตเป็นเท่าไหร่ ซึ่งตัวที่ทำหน้าที่เปรียบเทียบนี้มีชื่อว่า **coordinates** ผ่านเงื่อนไขการทำงานชื่อ **writeroute** และเมื่อทำการเปรียบเทียบหาค่าได้แล้วก็จะเขียนลงในไฟล์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* write route to file name "ddepth.txt" */
writeroute:-
        visited(A),
        coordinates(A, C, D),
        write(C), write(","), write(D), nl,
        fail,
        !.

coordinates(a, 0.25, 1).
coordinates(b, 1.256, 2).
coordinates(c, 1.25, 3).
coordinates(d, 2.256, 4).
coordinates(e, 4.25, 5).
coordinates(f, 8.256, 6).
coordinates(g, 5.25, 7).
coordinates(h, 2.256, 8).
coordinates(i, 2.25, 9).
coordinates(j, 7.256, 10).
coordinates(k, 9.25, 11).
coordinates(l, 3.256, 12).
coordinates(m, 6.25, 13).
coordinates(n, 4.256, 14).
coordinates(o, 10.25, 15).
```

6. ส่วนตัว Source code ทั้งหมดจะอยู่ในหนังสือ Souce Code ต่างหาก
7. ผลลัพธ์ที่ได้จากการใช้โปรแกรมการค้นหาเส้นทางโดยการค้นหาแบบตามแนวลึกก่อนนี้จะได้ผลดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ถ้าเราสมมติให้ว่าการเดินทางจากเมือง c ไปยังเมือง s ผลลัพธ์ที่ได้จากโปรแกรมมีดังนี้

```
From : c
To : s
Distance is 3013
Route is :
c
d
p
e
d
e
p
h
s
```

- ตัวอย่างการทำงานของโปรแกรมโดยเดินทางจากเมือง c ไปเมือง h คือ

```
From : c
To : h
Distance is 1350
Route is :
c
d
p
h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การเขียนโปรแกรมการค้นหาแบบตามแนวกว้างก่อน (Breadth-First Search)

การค้นหาแบบตามแนวกว้างก่อนนี้เป็นการค้นหาที่ตรงข้ามกับการค้นหาตามแนวลึกก่อน โดยการตรวจสอยโหนดที่อยู่ในระดับเดียวกันจนหมดก่อนจึงจะตรวจสอบโหนดที่อยู่ในระดับที่ลึกลงไป โดยทีขั้นตอนการเขียนโปรแกรมดังต่อไปนี้

1. การเขียนโปรแกรมนี้เป็นการแปลงรูปแบบการเดินทางของการค้นหาแบบตามแนวลึกก่อน เพราะว่า หลักการทำงานโดยรวมแล้วก็จะคล้ายๆกัน แต่จะต่างกันตรงที่ส่วนของการทำงานค้นหาเส้นทาง ซึ่งก็คือ **data** โดยเราจะทำการเปลี่ยนแปลงส่วนนี้ให้เป็นดังต่อไปนี้

```
data(T, T2, D):-
        move_to(T, T2, D),
        add_to_route(T),
        retractall(move_to(T, T2, D)).
data(T, T2, D):-
        move_to(T, X, D2),
        move_to(X, T2, D3),
        add_to_route(T),
        retractall(move_to(T, X, D2)),
        add_to_route(X),
        retractall(move_to(X, T2, D3)),
        D = D2+D3.
data(T, T2, D):-
        move_to(T, X, D2),
        X <> T2,
        add_to_route(T),
        retractall(move_to(T, X, D2)),
        data(X, T2, D3),
        D = D2+D3.
data(T, _, D):-
        write("Dead end at ", T),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nl,
D = 0,
fail.
```

ซึ่งเราจะเห็นความแตกต่างอย่างชัดเจนในส่วนที่เป็น

```
data(T, T2, D):-
        move_to(T, X, D2),
        move_to(X, T2, D3),
        add_to_route(T),
        retractall(move_to(T, X, D2)),
        add_to_route(X),
        retractall(move_to(X, T2, D3)),
        D = D2+D3.
```

เนื่องจากเงื่อนไขนี้จะทำงาน ถ้าเงื่อนไขแรกเป็นเท็จข้อมูลต่างๆที่เป็นอินพุทของเงื่อนไขที่หนึ่งก็จะเป็น ข้อมูลอินพุทของเงื่อนไขที่สอง ซึ่งเราก็รู้อยู่แล้วว่าไม่มีเส้นทางการเชื่อมต่อจากเมืองที่อยู่ในตัวแปร T ไป ยังเมืองที่อยู่ในตัวแปร T2 อยู่แล้ว (เพราะถ้ามี ก็จะไม่สามารถผ่านเงื่อนไขที่หนึ่งลงมาได้ ต้องทำงานเงื่อนไขที่หนึ่งไปแล้ว) ดังนั้นเงื่อนไขที่สองจะตรวจสอบว่ามีเมืองอื่นใดที่สามารถเชื่อมต่อกับเมืองที่อยู่ในตัวแปร T ได้หรือไม่ ถ้ามี เงื่อนไขก็จะตรวจสอบว่าเมืองนั้นเมืองจุดหมายที่ต้องการหรือไม่ แต่ถ้าไม่มีโปรแกรมก็จะย้อนรอยเพื่อไปตรวจสอบเมืองอื่นๆว่ามีเมืองไหนเชื่อมต่อจากเมืองนี้อีกหรือไม่ จนกว่าจะพบเมืองที่เป็นเมืองจุดหมายที่ต้องการหรือไม่ก็ไม่มีเมืองใดอีกแล้วที่เชื่อมต่อกับเมืองที่อยู่ในตัวแปร T นั้น และถ้าเงื่อนไขที่สองไม่สามารถค้นหาเส้นทางที่ไปยังเมืองจุดหมายที่ต้องการได้ ก็จะส่งต่อไปให้เงื่อนไขที่สาม ทำหน้าที่ค้นหาในระดับที่ลึกกว่านั้นลงไปเรื่อยๆ

2. แต่การทำงานของโปรแกรมนี้เราต้องการให้สามารถเขียนค่าโคออร์ดิเนตของแต่ละโหนดที่เป็นเส้นทางต่างๆลงในไฟล์ได้ ดังนั้นเราต้องทำการเปิดไฟล์เพื่อทำการเขียนลงไฟล์โดยสามารถทำได้โดยการใช้คำสั่งต่อไปนี้

```
openwrite(filetype, "dbread.txt"),
writedevice(filetype),
not(writeroute),
coordinates(B, C, D),
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
write(C), write(","), write(D),
closefile(filetype),
writedevice(screen),
not(displayroute),
write(B).
```

โดยทำการเก็บลงในไฟล์ชื่อ dbread.txt และในการเขียนลงในไฟล์ได้เราต้ องมีการค้นหาว่าโหนดนั้นมีค่าโคออร์ดิเนตเป็นเท่าไหร่ ซึ่งตัวที่ทำหน้าที่เปรียบเทียบนี้มีชื่อว่า **coordinates** ผ่านเงื่อนไขการทำงานชื่อ **writeroute** และเมื่อทำการเปรียบเทียบหาค่าได้แล้วก็จะเขียนลงในไฟล์ทั้งหมด

3. ผลลัพธ์ที่ได้จากการใช้โปรแกรมการค้นหาเส้นทางโดยใช้การค้นหาแบบตามแนวกว้างก่อนจะได้ผลดังต่อไปนี้

- ถ้าเราสมมติให้ว่าการเดินทางจากเมือง c ไปยังเมือง s ผลลัพธ์ที่ได้จากโปรแกรมมีดังนี้

```
From : c
To : s
Distance is 1569
Route is :
c
d
p
h
s
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวอย่างการทำงานของโปรแกรมโดยเดินทางจากเมือง c ไปเมือง h คือ

```
From : c
To : h
Distance is 1350
Route is :
c
d
p
h
```

3.3.3 การเขียนโปรแกรมการค้นหาเส้นทางแบบปีนตามเนินเขา

การค้นหาทั้งสองแบบก่อนหน้านี้ไม่ว่าจะเป็นการค้นหาแบบตามแนวลึกก่อน หรือการค้นหาตามแบบแนวกว้างก่อน เป็นการค้นหาแบบไม่มี Heuristic ทำให้ข้อมูลที่ได้ไม่สามารถรับประกันได้ว่าเป็นการเดินทางที่สั้นที่สุด (minimize) จริงหรือไม่ แต่คำว่า minimize สามารถพิจารณาออกได้เป็นสองแบบ นั่นคือ

1. จำนวนของจุดที่เชื่อมต่อในการเดินทางต้องน้อยที่สุด
2. ความยาวของเส้นทางน้อยที่สุด

ก่อนอื่นเราต้องรู้ว่าเส้นทางที่สั้นที่สุดไม่จำเป็นว่า จะต้องเป็นเส้นทางที่จุดโหนดที่ผ่านน้อยที่สุด อัลกอริทึมในการค้นหาพยายามที่จะหาผลลัพธ์ที่ดีที่สุดซักอันหนึ่งที่มีจำนวนจุดเชื่อมต่อที่น้อยที่สุดโดยการใช้ข้อมูลทาง Heuristic มาประกอบการตัดสินใจในการเลือกเส้นทางต่อๆไป เพื่อที่จะให้สามารถเข้าใกล้จุดหมายให้ได้มากที่สุด โดยพยายามลดจำนวนจุดที่เชื่อมต่อให้มากที่สุด

โดยปกติแล้วอัลกอริทึมในการค้นหาแบบปีนตามเนินเขาจะทำการเลือกโหนดที่เป็นขั้นตอนต่อไปที่สามารถเข้าไปใกล้จุดหมายให้มากที่สุด และ เมื่อประยุกต์เข้ากับการค้นหาเส้นทางระหว่างเมือง สิ่งที่เราต้องสนใจก็คือข้อมูลในการตัดสินใจว่าจะไปในเส้นทางใดดี โดยการใช้ Heuristic ของโปรแกรมในการตัดสินเส้นทางเริ่มต้นจากการเลือกเส้นทางการเชื่อมต่อที่สามารถทำให้เราไปจากจุดที่อยู่ปัจจุบันได้ไกลที่สุด โดยหวังว่าเส้นทางที่เราเลือกนั้น (ไปยังจุดที่อยู่ไกลที่สุด) จะทำให้เราอยู่ใกล้จุดหมายที่เราต้องการมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมนั้น เราสามารถนำโปรแกรมที่มีอยู่แล้วนำมาแก้ไขได้ โดยทำการแก้ไขในส่วนที่เป็น data และเพิ่มเงื่อนไขอีกอันหนึ่งเข้าไป นั่นก็คือ findlargest ซึ่งมีสามารถเขียนโปรแกรมได้ดังต่อไปนี้

```
data(T, T2, D):-
        move_to(T, T2, D),
        add_to_route(T),
        retractall(move_to(T, T2, D)).
data(T, T2, D):- /* climb hill */
        findlargest(T, X),
        add_to_route(T),
        move_to(T, X, D2),
        retractall(move_to(T, X, D2)),
        data(X, T2, D3),
        D = D2+D3.
data(T, T2, D):-
        move_to(T, X, D2),
        X <> T2,
        add_to_route(T),
        retractall(move_to(T, X, D2)),
        data(X, T2, D3),
        D = D2+D3.

findlargest(A, B):-
        move_to(A, X, D),
        move_to(A, Y, D2),
        X <> Y,
        D2 > D,
        B = Y.
```

โดยการเปลี่ยนแปลงที่เกิดขึ้น จะเกิดดังต่อไปนี้

```
data(T, T2, D):- /* climb hill */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
findlargest(T, X),
add_to_route(T),
move_to(T, X, D2),
retractall(move_to(T, X, D2)),
data(X, T2, D3),
D = D2+D3.
```

โดยส่วนที่ทำการแก้ไขจะทำงานดังต่อไปนี้คือ

1. findlargest พยายามจะหาเส้นทางระหว่างเมืองที่อยู่ไกลที่สุดระหว่างเมืองที่อยู่ในตัวแปร T และเมืองที่อยู่ในตัวแปร X
2. ถ้า findlargest ทำงานได้สำเร็จแล้ว มันก็จะเพิ่มเมือง T เข้าไปในเส้นทางและใช้การเพิ่ม flight(T, X, D2) เข้าไปเพื่อทำหน้าที่หาระยะห่าง D2
3. ขั้นตอนสุดท้าย ก็จะให้เงื่อนไข data ทำการเรียกตนเอง แต่ขอให้สังเกตว่า findlargest ทำให้โปรแกรมเกิดการย้อนรอยเพื่อที่จะหาเมืองที่อยู่ไกลที่สุดจากเมืองที่อยู่ในปัจจุบันนี้

แต่การทำงานของโปรแกรมนี้เราต้องการให้สามารถเขียนค่าโคออร์ดิเนตของแต่ละโหนดที่เป็นเส้นทางต่างๆลงในไฟล์ได้ ดังนั้นเราต้องทำการเปิดไฟล์เพื่อทำการเขียนลงไฟล์โดยสามารถทำได้โดยการใช้คำสั่งต่อไปนี้

```
openwrite(filetype, "dhill.txt"),
writedevice(filetype),
not(writeroute),
coordinates(B, C),
write(C),
closefile(filetype),
writedevice(screen),
not(displayroute),
write(B).
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทำการเก็บลงในไฟล์ชื่อ dhill.txt และในการเขียนลงในไฟล์ได้เราต้องมีการค้นหาว่าโหนดนั้นมีค่าโคออร์ดิเนตเป็นเท่าไหร่ ซึ่งตัวที่ทำหน้าที่เปรียบเทียบนี้มีชื่อว่า **coordinates** ผ่านเงื่อนไขการทำงานชื่อ **writeroute** และเมื่อทำการเปรียบเทียบหาค่าได้แล้วก็จะเขียนลงในไฟล์ทั้งหมด (ตัว Source Code อยู่ในหนังสือ Souce Code ต่างหาก)

ผลจากการทำงานของโปรแกรมการค้นหาเส้นทางโดยใช้การค้นหาแบบปีนตามเนินเขานี้ เราจะได้ว่า

- ถ้าเราสมมติให้ว่าการเดินทางจากเมือง c ไปยังเมือง s ผลลัพธ์ที่ได้จากโปรแกรมมีดังนี้

```
From : c
To : s
Distance is 6191
Route is :
c
o
d
o
c
b
o
l
p
h
s
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวอย่างการทำงานของโปรแกรมโดยเดินทางจากเมือง c ไปเมือง h คือ

```
From : c
To : h
Distance is 5972
Route is :
c
o
d
o
c
b
o
l
p
h
```

3.3.4 การค้นหาเส้นทางโดยใช้การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด

การค้นหาแบบนี้เป็นการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด ซึ่งเป็นวิธีที่ตรงข้ามกับวิธีการค้นหาแบบปีนตามเนินเขา โดยการค้นหาแบบนี้จะเน้นที่ว่าพยายามให้เสียค่าใช้จ่ายให้น้อยที่สุดเพื่อที่จะให้ได้เส้นทางที่เดินทางโดยใช้ความพยายามน้อยที่สุด

การประยุกต์การค้นหาแบบก่อนๆ มาใช้กับการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด เพื่อใช้ในการค้นหาเส้นทาง สามารถพูดได้ว่าเส้นทางที่เชื่อมต่อที่ได้จะเป็นเส้นทางการเดินทางที่สั้นที่สุด ซึ่งตรงข้ามกับการค้นหาแบบปีนตามเนินเขา ที่จะพยายามให้มีจำนวนโหนดที่เดินทางผ่านให้น้อยที่สุด แต่การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุดจะพยายามให้มีเส้นทางการเดินทางให้สั้นที่สุด

การเขียนโปรแกรมการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด เราต้องปรับเปลี่ยนส่วนของเงื่อนไข data ให้เป็นไปดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
data(T, T2, D):-
        move_to(T, T2, D),
        add_to_route(T),
        retractall(move_to(T, T2, D)).
data(T, T2, D):- /* find shortest */
        findsmallest(T, X),
        add_to_route(T),
        move_to(T, X, D2),
        retractall(move_to(T, X, D2)),
        data(X, T2, D3),
        D = D2+D3.
data(T, T2, D):-
        move_to(T, X, D2),
        X <> T2,
        add_to_route(T),
        retractall(move_to(T, X, D2)),
        data(X, T2, D3),
        D = D2+D3.

findsmallest(A, B):-
        move_to(A, X, D),
        move_to(A, Y, D2),
        X <> Y,
        D2 > D,
        B = Y.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราจะเห็นว่า สิ่งที่แตกต่างระหว่างการค้นหาแบบปีนตามเนินเขาและการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุดคือ ก็คือการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุดจะหาเส้นทางที่สั้นที่สุดในการเดินทางแต่ละครั้ง

แต่การทำงานของโปรแกรมนี้เราต้องการให้สามารถเขียนค่าโคออร์ดิเนตของแต่ละโหนดที่เป็นเส้นทางต่างๆลงในไฟล์ได้ ดังนั้นเราต้องทำการเปิดไฟล์เพื่อทำการเขียนลงไฟล์โดยสามารถทำได้โดยการใช้คำสั่งต่อไปนี้

```
openwrite(filetype, "dleast.txt"),
writedevice(filetype),
not(writeroute),
coordinates(B, C),
write(C),
closefile(filetype),
writedevice(screen),
not(displayroute),
write(B).
```

โดยทำการเก็บลงในไฟล์ชื่อ dleast.txt และในการเขียนลงในไฟล์ได้เราต้องมีการค้นหาว่าโหนดนั้นมีค่าโคออร์ดิเนตเป็นเท่าไหร่ ซึ่งตัวที่ทำหน้าที่เปรียบเทียบนี้มีชื่อว่า **coordinates** ผ่านเงื่อนไขการทำงานชื่อ **writeroute** และเมื่อทำการเปรียบเทียบหาค่าได้แล้วก็จะเขียนลงในไฟล์ทั้งหมด (ตัวโปรแกรมที่เป็น Source Code อยู่ในหนังสือ Source Code ต่างหาก)

ซึ่งผลที่ได้เป็นดังต่อไปนี้

- ถ้าเราสมมติให้ว่าการเดินทางจากเมือง c ไปยังเมือง s ผลลัพธ์ที่ได้จากโปรแกรมมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
From : c
To : s
Distance is 6191
Route is :
c
o
d
o
c
b
o
l
p
h
s
```

- ตัวอย่างการทำงานของโปรแกรมโดยเดินทางจากเมือง c ไปเมือง h คือ

```
From : c
To : h
Distance is 5972
Route is :
c
o
d
o
c
b
o
l
p
h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 สรุปผลการทำงานรวมของการค้นหาแบบวิธีต่างๆทั้ง 4 วิธีที่ผ่านมา

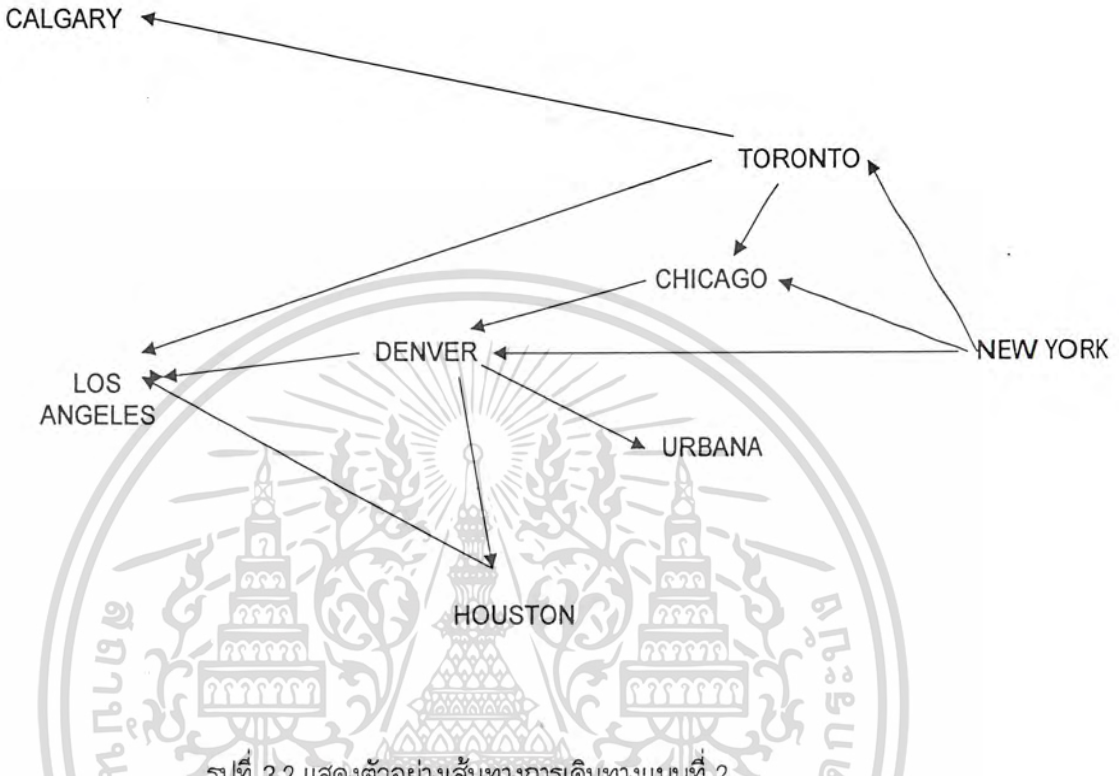
จากผลที่ผ่านมา เราสามารถสรุปผลได้ดังตารางต่อไปนี้

การค้นหา	c ไป h	p ไป h	c ไป s	p ไป q	p ไป r	o ไป j	c ไป k
Depth-First	5972	628	6191	311	6850	3362	4314
Breadth-First	1350	628	1569	311	929	624	1614
Hill-Climbing	5972	628	6191	311	6850	3362	5258
Least-Cost	5972	628	6191	311	6850	3362	5258

ตารางที่ 3.1 แสดงการเปรียบเทียบวิธีการค้นหาเส้นทางโดยใช้วิธีต่างๆกันกับตัวอย่างแบบที่ 1

จากตารางเราพบว่าวิธีการค้นหาแบบ Heuristic กลับไม่สามารถช่วยให้เส้นทางมีขนาดสั้นลงได้เลย แต่กลับว่าการค้นหาแบบไม่มี Heuristic มีความสามารถในการหาเส้นทางได้สั้นมากกว่า โดยเฉพาะการค้นหาแบบตามแนวกว้างก่อน สามารถหาเส้นทางได้สั้นที่สุด และสั้นกว่าเส้นทางอื่นมากในการหาบางเส้นทาง เหตุผลที่เป็นเช่นนี้เพราะว่าเรามีการจัดโครงสร้างฐานข้อมูล ที่สนับสนุนการค้นหาแบบกว้างก่อนได้เป็นอย่างดี แต่ถ้าเป็นกรณีที่ข้อมูลเป็นแบบอื่น เช่นสมมติเส้นทางการค้นหาดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงตัวอย่างเส้นทางการเดินทางแบบที่ 2

ผลลัพธ์ที่ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหา	New York ไป Denver	New York ไป Los Angeles	New York ไป Houston	New York ไป Calgary	Toronto ไป Los Angeles	Toronto ไป Urbana	Chicago ไป Houston
Depth-First	1900	3000	3500	1000	1800	6700	2500
Breadth-First	1900	2600	3400	2300	1800	6700	2500
Hill-Climbing	1900	2900	3400	1000	1800	5600	2500
Least-Cost	1900	2900	3400	1000	1800	5600	2500

ตารางที่ 3.2 แสดงการเปรียบเทียบวิธีการค้นหาเส้นทางโดยใช้วิธีต่างๆกันกับตัวอย่างแบบที่ 2

จากผลที่เราได้ เราจะเห็นว่าผลที่ได้จากการค้นหาแบบ Heuristic เป็นวิธีที่ได้ผลลัพธ์ดีที่สุด ซึ่งตรงกับทฤษฎีการใช้ Heuristic จากวิธีการดังกล่าวเกิดจากการจัดฐานข้อมูลที่ไม่เป็นระเบียบทำให้ผลที่ได้จึงเหมาะสมกับการใช้วิธีการค้นหาแบบปีนภูเขา หรือวิธีการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด

ทำไมผลที่ได้จากการตัวอย่างทั้งสองจึงสรุปได้ไม่เหมือนกัน

ที่เกิดขึ้นเช่นนี้เนื่องจากว่าในตัวอย่างแรกเป็นตัวอย่างที่มีฐานความรู้ที่ค่อนข้างจะถูกจัดให้มีระเบียบ ทำให้การค้นหาแบบตามแนวกว้างก่อนหรือ การค้นหาแบบตามแนวลึกก่อนจึงทำงานได้อย่างมีประสิทธิภาพมากกว่า และการที่การค้นหาแบบตามแนวกว้างก่อนสามารถทำงานได้ดีที่สุดเนื่องจากว่าการจัดลำดับของฐานข้อมูลเหมาะสมกันการค้นหาแบบนี้พอดี ทำให้ผลที่ได้จึงเหมาะกับการทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโครงสร้างฐานข้อมูลของทั้งสองตัวอย่างมีความแตกต่างอย่างชัดเจน คือ ตัวอย่างแรกมีการโครงสร้างของฐานข้อมูลจัดอย่างมีระเบียบเรียบร้อยและการจัดจะจัดเรียงกันอย่างมีลำดับฐานข้อมูลอย่างต่อเนื่องโดยเรียงลำดับตามระยะห่างระหว่างโหนดเป็นหลัก แต่ในตัวอย่างที่สองมีการจัดลำดับการเรียงข้อมูลอย่างสะเปะสะปะทำให้การค้นหาแบบตามแนวลึกก่อนและการค้นหาแบบตามแนวกว้างก่อน อาจพบกับข้อมูลที่มักเป็นข้อมูลที่ไม่แน่ว่าจะเป็นเส้นทางที่น้อยที่สุดของโหนดนั้นหรือไม่ก็ได้ แต่ทว่าการทำงานแบบเสียค่าใช้จ่ายน้อยที่สุดและการค้นหาแบบปีนภูเขาเป็นการค้นหาเส้นทางที่สั้นที่สุดสำหรับการเดินทางเสมอจากโหนดเริ่มต้นนั้นๆ แต่ทว่าก็อาจไม่ใช่ข้อมูลที่น้อยที่สุดจริงๆ อย่างเช่น

move_to(a, b, 2)
move_to(a, g, 6)
move_to(a, c, 9)
move_to(a, h, 10) เป็นต้น

ซึ่งการเรียงแบบนี้การค้นหาแบบกว้างก่อนและการค้นหาแบบลึกก่อนนั้น สามารถค้นหาได้ดีเท่ากับการค้นหาแบบปีนภูเขาและการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด แต่ถ้าการเรียงลำดับเป็นแบบ

move_to(a, c, 9)
move_to(a, b, 2)
move_to(a, h, 10)
move_to(a, g, 6)

ถ้าเรียงลำดับฐานข้อมูลแบบนี้ก็จะทำให้การค้นหาแบบตามแนวลึกก่อนและการค้นหาแบบตามแนวกว้างก่อน ทำการค้นหาข้อมูลไม่สามารถทำได้อย่างมีประสิทธิภาพเหมือนแบบก่อนเพราะว่าไม่มีการเรียงลำดับการค้นหาข้อมูล ดังนั้นถ้าข้อมูลไม่ได้มีการจัดอย่างมีประสิทธิภาพควรใช้การค้นหาแบบมี Heuristic คือ แบบปีนตามภูเขาและการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด จะทำการค้นหาได้ดีกว่า แต่ถ้าเป็นกรณีการสลับข้อมูลเป็นแบบ

move_to(a, b, 12)
move_to(c, h, 5)
move_to(b, c, 6)
move_to(a, h, 10)
move_to(b, m, 9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งถ้าเป็นกรณีนี้การค้นหาแบบ Heuristic และการค้นหาแบบไม่มี Heuristic ก็มีความสามารถในการค้นหาที่ไม่แตกต่างกัน เพราะว่าการเรียงไม่มีการเรียงลำดับการค้นหาแบบใดๆเลย

3.4 การ Digitize ภาพแผนที่

3.4.1 การติดตั้งเครื่อง Digitizer

ก่อนอื่นต้องทำการติดตั้งเครื่อง digitizer ให้เหมาะสมกับงานที่ต้องทำก่อน โดยเครื่อง digitizer ที่ใช้งานนี้เป็นเครื่อง digitizer ขนาด 12"X12" ของ GENIUS ซึ่งมีขั้นตอนการติดตั้งดังนี้

1. พิมพ์คำสั่ง Config ที่ Command Line แล้วหน้าจอจะเปลี่ยนเป็นแสดงผลสถานะของการติดตั้งปัจจุบัน ว่าได้เลือกที่จะใช้อะไร เราก็กด key ใดๆ เพื่อเข้าสู่การติดตั้ง
2. ทำการเลือกหมายเลข 4 เพื่อเลือกการติดตั้ง digitizer แล้วให้ตอบว่า Y เพื่อยืนยันการเปลี่ยนแปลงโดยจะบอกว่าตอนนี้ใช้อะไรอยู่ต้องการเปลี่ยนแปลงหรือไม่
3. เลือกประเภทหมายเลข 11 แล้วกด Enter ตั้งขนาดเป็นหมายเลข 3 แล้วกด Enter
4. เลือกประเภท 1 เพื่อตั้งให้การทำงานเกิดขึ้นในลักษณะ Interrupt Driven แล้วเลือกประเภทจำนวนปุ่มว่า เครื่องที่เราใช้มี mouse ที่มีการทำงานอยู่กี่ปุ่ม
5. เลือกว่า digitizer ของเราต่อเข้ากับ port ใด เช่น comm1, comm2, lpt1, lpt2 เป็นต้น
6. กลับไปสู่ส่วน menu หลักเพื่อเลือกการทำงานว่าต้องการติดตั้งอุปกรณ์ประเภทอื่นอีกหรือไม่ ถ้าต้องการก็ทำการเลือกหมายเลขนั้น แต่ถ้าติดตั้งเสร็จแล้วก็เลือกหมายเลข 0 เพื่อกลับไปสู่หน้าจอที่เป็น graphic แล้วเครื่องจะถามว่าต้องการ save ส่วนที่ทำการติดตั้งใหม่หรือไม่ ถ้าเราตอบ Y สิ่งต่างๆที่เราได้ติดตั้งก็จะมีผล แต่ถ้าเลือก N สิ่งต่างๆที่ได้ติดตั้งมาจะไม่มีผลใดๆ เลย

การติดตั้งอุปกรณ์ใดๆ ควรดูที่คู่มือที่แนบมากับอุปกรณ์เพราะว่าอุปกรณ์แต่ละชนิดก็จะมีการติดตั้งที่ต่างกัน ที่ยกตัวอย่างมานี่เป็นแค่การติดตั้งอุปกรณ์ที่ใช้ในการ digitize ในที่นี้เท่านั้น

และเมื่อทำการติดตั้ง digitizer เสร็จแล้วก็ทำการติดตั้ง digitizer ซึ่งก็คือ tablet ในที่ที่ AutoCAD รู้จักนั่นเอง โดยทำการติดตั้งดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. พิมพ์คำสั่ง tablet ที่ Command Line แล้วกด Enter

2. พิมพ์ cfg แล้วกด Enter

3. เลือกจำนวนของพื้นที่ที่ต้องการแบ่งใน digitizer โดยมีให้เลือกตั้งแต่ <0-4> แล้วก็กำหนดขอบเขตแต่ละพื้นที่ (การติดตั้งในส่วนนี้ขอให้เลือกพื้นที่เป็น 0 เพราะว่าเราต้องการพื้นที่ในการ digitize มาก แล้วกำหนดขอบเขตให้ครอบคลุมทั้งแผ่น digitize

ทั้งหมดนี้เป็นกระบวนการติดตั้งเครื่อง digitizer ให้เหมาะกับการทำงาน ขั้นตอนต่อไปเป็นการปรับลักษณะของ digitizer ให้ตรงกับลักษณะงานที่เราใช้ โดยงานที่เราใช้เป็นงานแผนที่ ทำให้ลักษณะของงานเป็นประเภทคัดลอกแผนที่ลงในเครื่องคอมพิวเตอร์ ซึ่งมาตราส่วนของแผนที่ที่ใช้จึงมีส่วนสำคัญ ดังนั้นก่อนที่เราจะนำแผนที่ไปวางบนเครื่อง digitizer เพื่อทำการคัดลอก เราต้องนำแผนที่นั้นมาตีเส้นเป็นตารางสี่เหลี่ยมก่อน เพื่อที่จะสามารถเปรียบเทียบมาตราส่วนที่เห็นในแผนที่ 1 ช่องเท่ากับมาตราส่วนที่เห็นในหน้าจอ 1 ช่องเช่นกัน ดังนั้นถ้าเราตีเส้นให้มีความถี่มากเพียงใด เราก็จะได้ภาพที่มีความละเอียดมากขึ้นเท่านั้น แต่ในที่นี้เราจะตีให้เส้นแต่ละเส้นห่างกัน 1 นิ้ว

หลังจากนั้นก็นำแผนที่นั้นไปวางบนเครื่อง digitizer แล้วเอาเทปใสมาติดกระดาษ แผนที่ให้ติดกับเครื่อง digitizer แล้วกำหนดสิ่งต่างๆดังต่อไปนี้

1. ที่ Command Line ให้พิมพ์ LIMITS <กด Enter> แล้วตั้งค่าไว้ที่ 180, 180 <กด Enter>

2. ที่ Command Line ให้พิมพ์ GRID <กด Enter> เลือก ON แล้วกด Enter

3. ที่ Command Line ให้พิมพ์ TABLET กด Enter แล้วเลือก CAL แล้วเครื่องจะถามถึงจุดที่ต้องการตั้งค่าแผนที่ เราก็นำ mouse ที่ชี้ที่มุมซ้ายบนในหน้าจอ แล้วเลื่อนไปชี้ที่มุมล่างขวาของตารางนั้นในแผนที่ แล้วไปชี้ที่มุมล่างขวาของตารางเดิมในหน้าจอ แล้ว Click

3.4.2 การใช้งานเครื่อง Digitizer

1. เราสามารถลอกภาพแผนที่ได้โดยใช้คำสั่งต่างๆ เช่น LINE, ARC, ERASE, MOVE ฯลฯ โดยใช้คำสั่งเหล่านี้ จะทำงานที่ command line : ทั้งสิ้น
2. เมื่อเสร็จจากหน้าแรกแล้ว เราก็เลื่อนแผนที่ส่วนที่เหลือเข้ามาแล้วติดเทปใสใหม่ไปเรื่อยๆ (ข้อควรระวังในการคัดลอกแผนที่ ห้ามให้แผ่นแผนที่เลื่อนเพราะจะทำให้ต้องตั้ง config กันใหม่ และ scale ที่ได้อาจผิดพลาดด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าต้องการเลื่อนตำแหน่งหน้าจอ ก็ใช้คำสั่ง PAN เช่นถ้าต้องการเลื่อนหน้าจอไปทางด้านขวามือ ก็พิมพ์ที่ command line ว่า PAN แล้วไป click ที่ด้านขวาของจอภาพปัจจุบัน แล้วกดค้าง ลากไปทางด้านซ้าย แล้ว click อีกทีหน้าจอก็จะเลื่อนไป โดยการที่หน้าจอจะเลื่อนมากหรือน้อยขึ้นอยู่กับ การลากปุ่มของเรา ถ้าลากมากก็เลื่อนมาก เป็นต้น
4. ควร save เป็นระยะๆ เพื่อป้องกันมิให้ข้อมูลหาย โดยพิมพ์ที่ command line : ว่า save <Enter>แล้วเครื่องจะป้อน ชื่อ file ที่จะ save ลงไป

ผลลัพธ์ที่ได้จากการ digitize ภาพแผนที่ถนนเยาวราชได้ดังรูปในภาคผนวก ง

3.5 การรวมเอาส่วนโปรแกรมและเอาส่วนของภาพแผนที่เข้ามาประกอบกัน

เมื่อเราได้ภาพแผนที่แล้วสิ่งสำคัญที่เราต้องรู้คือ จุดโคออร์ดิเนตของโหนดต่างๆ ทั้งโหนดที่เป็น สี่แยก สามแยก โค้งหักมุมถนน โดยมีขั้นตอน ดังนี้

1. กำหนดชื่อสี่แยก สามแยก โค้งหักมุมถนน ต่างๆ ให้มีชื่อเป็นรหัสที่เราเข้าใจ แต่เราจะให้เฉพาะชื่อสี่แยก สามแยกเท่านั้น ที่ปรากฏในหน้าจอได้

2.หาจุดกึ่งกลางของสี่แยกนั้น หาค่าโคออร์ดิเนต แล้วบันทึกเอาไว้

3.หาว่าโหนดไหนเชื่อมต่อกับโหนดไหนบ้าง เช่น กำหนด (A, B, 10) หมายถึง เส้นทางเดินรถจาก จุด A ไปจุด B มีค่า weight เป็น 10 โดยถ้ามีการเดินรถจาก B ไป A ได้ก็ต้องเป็น (B, A, 10) เป็นต้น

4. หาค่า weight ของแต่ละช่วงโหนด แล้วบันทึกไว้

5. เอาค่าที่บันทึกไว้มาใส่ในโปรแกรมค้นหาภาพแผนที่ต่างๆ

6. หาเส้นทางแล้วเก็บลงใน text file แล้วทำการเรียก AutoCAD มาวาดเป็นเส้นได้

เราจะมาให้รายละเอียดแต่ละขั้นตอนดังนี้

ขั้นตอนที่ 1 กำหนดชื่อโหนดต่างๆ ได้โดยการเอาค่า low กับค่า column มาต่อกัน 2 โหนดใน 1 ช่องตาราง ก็กำหนดเลขท้ายสุดเป็น 1ไปเรื่อยๆ

ขั้นตอนที่ 2 หาค่าโคออร์ดิเนตของจุดกึ่งกลางของโหนด ได้ดังนี้ แสดงรายชื่อโหนด ค่าโคออร์ดิเนตในภาคผนวก ค

ขั้นตอนที่ 3 หาว่าโหนดไหนเชื่อมต่อกับโหนดไหนบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4 หาค่า weight ของเส้นทางระหว่างโหนดต่างๆที่อยู่ติดกัน (adjacent)

ขั้นตอนที่ 5 เอาข้อมูลที่ได้ไปใส่ในตัวโปรแกรม การค้นหาเส้นทางทั้ง 4 โดยมีลักษณะดังนี้

(ดูที่ source code)

(move_to (A, B, __))

(coordinate (__, __,__))

ขั้นตอนที่ 6 เปิดโปรแกรม AutoCAD แล้วพิมพ์ที่ command line เพื่อ load โปรแกรมการวาดเส้นทางที่ชื่อ path.lsp (Source code อยู่ที่ภาคผนวก) และเป็นภาพแผนที่ โดยกำหนด layer เอาไว้ และใช้คำสั่ง (data "Filename) <Enter> โดยที่ Filename คือ ชื่อ File ที่เป็นผลลัพธ์ของแต่ละวิธีการค้นหา

ซึ่งมีชื่อ File ดังนี้ ddepth.txt เป็นผลลัพธ์ของการค้นหาแบบ Depth-First Search

และ dbread.txt เป็นผลลัพธ์ของการค้นหาแบบ Breadth-First Search

และ dhill.txt เป็นผลลัพธ์ของการค้นหาแบบ Hill-Climbling Search

และ dleast.txt เป็นผลลัพธ์ของการค้นหาแบบ Least-Cost Search

สรุปขั้นตอนการใช้งาน

1. เปิด AutoCAD แล้วเปิดภาพแผนที่ถนนเยาวราชขึ้นมา
2. พิมพ์ shell <Enter> 2 ครั้ง
3. เรียกโปรแกรมค้นหาเส้นทางแบบที่ต้องการ ป้อนจุดเริ่มต้นและจุดปลาย ก็จะได้รายชื่อโหนดแสดงบนหน้าจอ และค่าโคออร์ดิเนตเก็บใน File ต่างๆที่เป็นผลลัพธ์ของแต่ละวิธีการค้นหา
4. พิมพ์ exit ที่ dos prompt
5. เรียก (load "path.lsp")
6. เรียก (data "Filename")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการวิเคราะห์

เนื่องจากการค้นหาเส้นทางแผนที่ทั้งแผ่นจำเป็นต้องใช้หน่วยความจำจำนวนมาก แต่เครื่องที่ใช้มีขนาดหน่วยความจำแค่ 8 MB ทำให้ขนาดแผนที่จำเป็นต้องถูกตัดออกไป โดยขอบเขตของแผนที่ที่จะนำมาใช้กับโปรแกรมนี้จะแคบลงเหลือแค่ 1 ใน 8 ส่วนของแผนที่ทั้งหมดที่ทำการเก็บ แต่ขั้นตอนการทำงานของโปรแกรมก็มีส่วนคล้ายกันดังนี้

1. เข้าโปรแกรม AutoCAD

```
C:\>ACADR12          <Enter>
```

2. เรียกภาพเส้นทางถนนเยาวราชออกมาก่อน

```
Command : open       <Enter>
                     <เลือกไฟล์ proj.dwg>
                     <OK>
```

3. ออกจากโปรแกรม AutoCAD ทาง Shell เพื่อเรียกโปรแกรมภายนอก โดยใช้คำสั่ง

```
Command : shell      <Enter>
```

4. ใช้โปรแกรมภายนอกที่เขียนจากภาษา Prolog มาหาเส้นทางโดยส่งผลลัพธ์ทางไฟล์ซึ่งเป็นข้อมูลในรูปของค่าโคออร์ดิเนตของเส้นทางที่หาได้นั้น

โดย ถ้าต้องการใช้โปรแกรมการค้นหาเส้นทางแบบตามแนวลึกก่อน ก็เรียกโปรแกรม depth.exe
ถ้าต้องการใช้โปรแกรมการค้นหาเส้นทางแบบตามแนวกว้างก่อน ก็เรียกโปรแกรม breadth.exe
ถ้าต้องการใช้โปรแกรมการค้นหาเส้นทางแบบปีนตามเนินเขา ก็เรียกโปรแกรม hill.exe
ถ้าต้องการใช้โปรแกรมการค้นหาเส้นทางแบบเสียค่าใช้จ่ายน้อยที่สุด ก็เรียกโปรแกรม least.exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เราจะต้องป้อนโหนดเริ่มต้นและโหนดปลายทางที่เราต้องการ
6. เครื่องจะแสดงเส้นทางเป็นจุดโหนดต้นทาง ผ่านโหนดต่างๆ ไปยังโหนดปลายทาง
7. กลับเข้าไปสู่ AutoCAD อีกครั้งหนึ่ง โดยการพิมพ์ EXIT <ENTER>

```
Command : exit                    <Enter>
```

8. ใช้คำสั่ง (load "path.lsp")

```
Command : (load "path.lsp")
```

9. เรียกไฟล์ที่ต้องการแสดงผล

โดย ไฟล์ ddepth.txt เป็นผลลัพธ์ของการค้นหาของไฟล์ depth.exeใช้คำสั่ง (data "ddepth.txt")
ไฟล์ dbread.txt เป็นผลลัพธ์ของการค้นหาของไฟล์ breadth.exeใช้คำสั่ง (data "dbread.txt")
ไฟล์ dhill.txt เป็นผลลัพธ์ของการค้นหาของไฟล์ hill.exeใช้คำสั่ง (data "dhill.txt")
ไฟล์ dleast.txt เป็นผลลัพธ์ของการค้นหาของไฟล์ least.exeใช้คำสั่ง (data "dleast.txt")
เช่น ถ้าต้องการให้แสดงผลของการค้นหาจากไฟล์ depth.exe ก็พิมพ์

```
Command : (data "ddepth.txt")
```

10. เครื่องก็จะสร้างเส้นทางให้เห็นออกทางหน้าจอภาพเป็นอีกเส้นทางดังกล่าวจะแสดงเป็นสีเหลือง
11. ผลที่ได้จะเป็นไปดังภาคผนวก ง

แผนที่ที่นำมาศึกษาถูกแสดงในภาคผนวก ค ส่วนผลการทำงานของแผนที่ทั้งแผ่นโดยใช้เครื่องที่มีขนาดหน่วยความจำสูงจะแสดงดังภาคผนวก ง

ในการวัดระยะทาง เราได้ทำการวัดให้มีหน่วยเป็นนิ้ว เนื่องจากหน่วยของค่า weight ที่ต้องการเป็นหน่วยวินาทีดังนั้นจึงต้องมีเปลี่ยนแปลงสูตรการคำนวณเพื่อให้เป็นในลักษณะที่ต้องการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

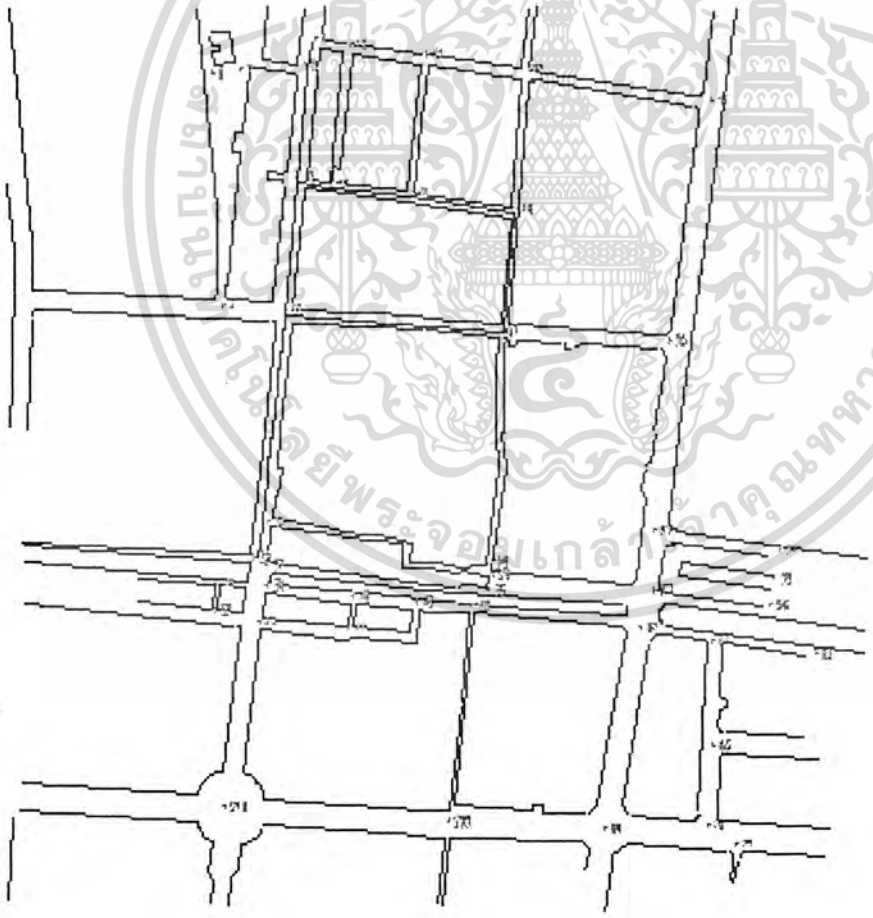
T (วินาที) = (25/12) * (S * W)

เมื่อ S มีหน่วยเป็นนิ้ว

W เป็นค่า weight ที่ไม่มีหน่วย

4.1 ผลที่ได้จากการใช้การค้นหาแบบตามแนวลึกก่อน

เมื่อเราทดสอบการทำงานโดยให้เดินทางจากโหนดหมายเลขที่ 19 ไปยังโหนดหมายเลข 38 จะได้ผลดังรูปที่ 4.1

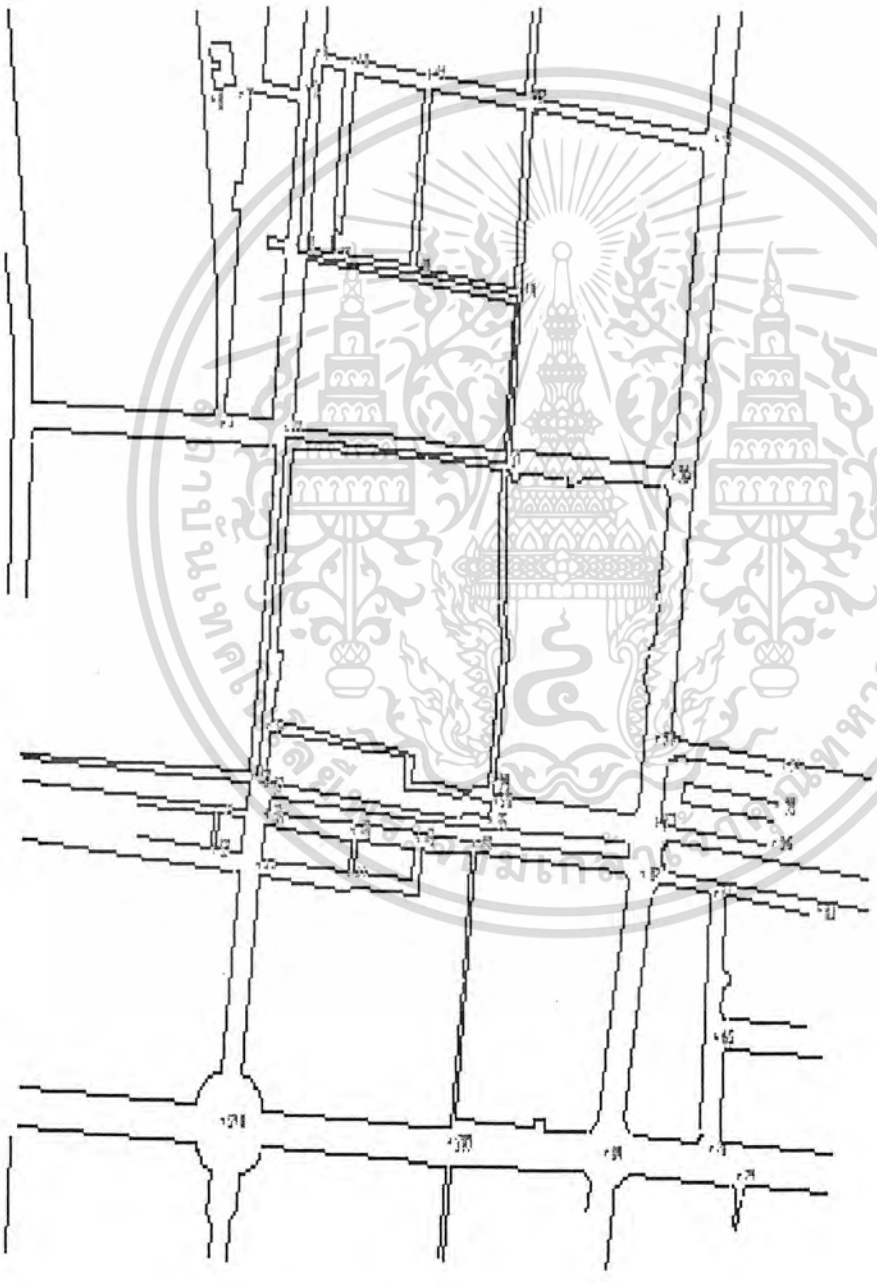


รูปที่ 4.1 แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบตามแนวลึกก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลที่ได้จากการใช้การค้นหาแบบตามแนวกว้างก่อน

เมื่อเราทดสอบการทำงานโดยให้เดินทางจากโหนดหมายเลขที่ 19 ไปยังโหนดหมายเลข 38 จะได้ผลดังรูปที่ 4.2

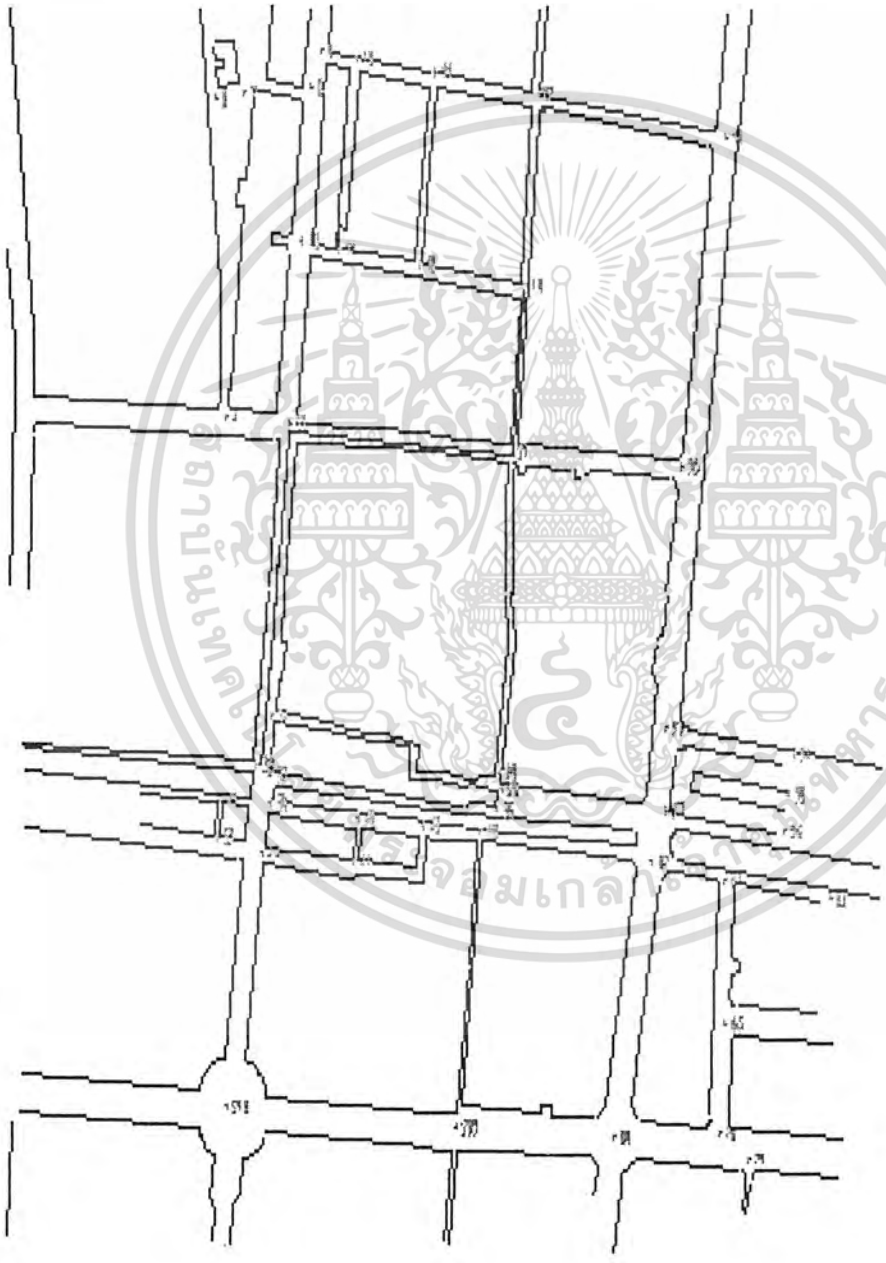


รูปที่ 4.2 แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบตามแนวกว้างก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลที่ได้จากการใช้การค้นหาแบบปีนตามเนินเขา

เมื่อเราทดสอบการทำงานโดยให้เดินทางจากโหนดหมายเลขที่ 19 ไปยังโหนดหมายเลข 38 จะได้ผล ดังรูปที่ 4.3

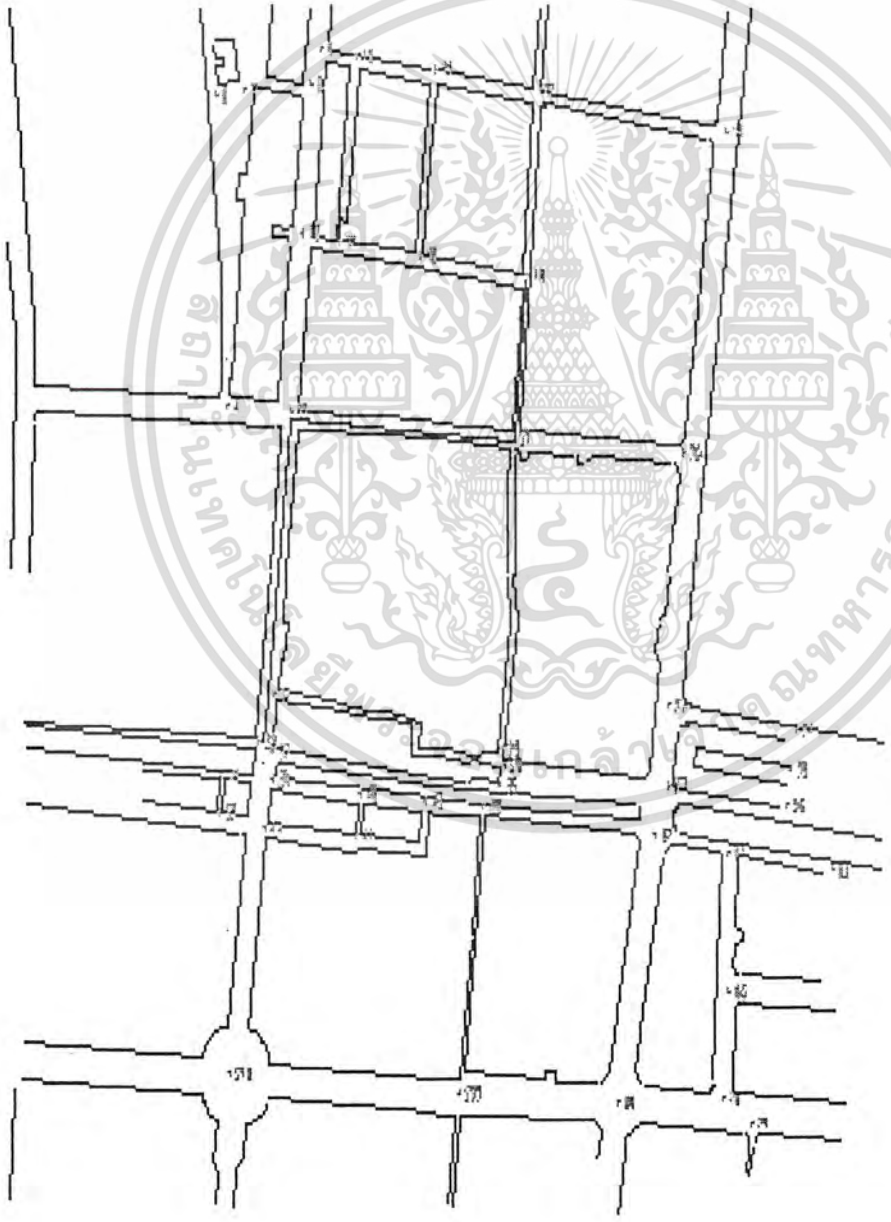


รูปที่ 4.3 แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบปีนตามเนินเขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลที่ได้จากการใช้การค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด

เมื่อเราทดสอบการทำงานโดยให้เดินทางจากโหนดหมายเลขที่ 19 ไปยังโหนดหมายเลข 38 จะได้ผล ดังรูปที่ 4.4



รูปที่ 4.4 แสดงผลลัพธ์ของการค้นหาเส้นทางด้วยวิธีการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 สรุปผลการใช้งานโปรแกรมในการค้นหาเส้นทางวิธีต่างๆ

การค้นหาภาพแผนที่โดยใช้วิธีการต่างๆในที่นี้เราแบ่งประเภทของการค้นหาออกเป็น 2 ประเภทคือ การค้นหาแบบไม่มี heuristic (Depth-first search, Breadth-first search) และการค้นหาแบบมี heuristic (Hill-climbing, Least-Cost) ซึ่งสรุปผลได้ดังนี้

การค้นหา	ประเมิน
Depth-Frist Search	พอใช้
Breadth-First Search	พอใช้
Hill-Climbing Search	ดี
Least-Cost Search	ดี

ตารางที่ 4.1 แสดงผลการสรุปวิธีการค้นหาเส้นทางที่ใช้วิธีการต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิเคราะห์และข้อเสนอแนะ

5.1 สรุปผลการวิเคราะห์

จากผลการทดลองดังกล่าวทำให้สรุปได้ว่าการค้นหาภาพแผนที่การจราจรกรุงเทพมหานคร โดยใช้ฐานความรู้นั้นเป็นวิธีการค้นหาภาพการจราจรที่จะเชื่อถือได้หรือไม่ขึ้นอยู่กับค่าน้ำหนัก (weight) ที่เราป้อนเข้าไป เพราะว่าวิธีการค้นหาแบบ heuristic ซึ่งเป็นวิธีการค้นหาที่มีประสิทธิภาพสูงอยู่แล้วนั้นค้นหาตามเส้นทางโดยพิจารณาค่าน้ำหนักเป็นหลัก ส่วนการค้นหาแบบไม่มี heuristic นั้นไม่คำนึงถึงค่าน้ำหนักแต่จะเป็นการค้นหาแล้วแต่การจัดฐานข้อมูลของคนเขียนโปรแกรมเอง แต่ผลที่ได้ก็ไม่ค่อยจะได้เส้นทางที่ถูกต้องเหมือนกับการใช้วิธีทาง heuristic ดังนั้นการหาค่าน้ำหนักจึงเป็นสิ่งที่สำคัญมาก แต่การหาค่าน้ำหนักนั้นเป็นงานที่ค่อนข้างยาก อาจต้องใช้วิธีการต่างในพิจารณาหาค่าโดยประมาณออกมา เช่น การใช้ Neural Network เป็นต้น แต่ทว่าการใช้วิธีต่างๆในการเก็บรวบรวมข้อมูลก็เป็นงานที่ยากมาก อาจต้องออกแบบสอบถามเพื่อถามถึงระยะเวลาที่ใช้ในการเดินทางในถนนแต่ละเส้นของแต่ละวัน เป็นต้น แต่ก็ยังมีปัจจัยที่ไม่แน่นอนประกอบขึ้นอีกมากมาย เช่น ถ้าวันนั้นเป็นวันหยุดราชการ การพิจารณาค่าน้ำหนักของถนนแต่ละเส้นก็จะเปลี่ยนไปโดยบางเส้นอาจมีค่าน้ำหนักที่มากขึ้นแต่บางเส้นอาจมีค่าน้ำหนักที่น้อยลง หรือว่าถ้าวันนั้นถนนบางเส้นเกิดมีการซ่อมแซมถนน หรือเกิดอุบัติเหตุรถยนต์ในถนนสายนั้นๆ ก็จะทำให้ค่าน้ำหนักต้องมีการเปลี่ยนแปลงบางช่วงเวลา เป็นต้น ซึ่งกรณีอย่างนี้มีวิธีแก้ไขอยู่คือการตั้งสถานีรับส่งข้อมูลสภาพจราจรถนน แต่ทว่าวิธีนี้ก็มีปัญหาอีกมากมายเนื่องจากว่า แต่ละคนที่ส่งข้อมูลเข้ามาก็ไม่สามารถบอกได้ว่า รถที่ติดตอนนี้ติดมากแค่ไหนควรให้ค่าน้ำหนักเป็นอย่างไร เพราะแต่ละคนก็มองสภาพการจราจรต่างๆกัน ดังนั้นจึงต้องมีการอบรมผู้ที่จะรายงานข้อมูลเข้ามาอีก ซึ่งจะเห็นได้ว่าการจะให้ได้ผลที่ถูกต้องนั้นเป็นงานที่ยากลำบาก, ใช้เวลาในการดำเนินงานนานและใช้ต้นทุนในการดำเนินการสูง

นอกจากเรื่องดังกล่าวแล้ว ยังมีปัญหาอย่างอื่นอีกคือ

1. การเก็บภาพแผนที่

เนื่องจากว่าการดิจิไตซ์ลงเครื่องคอมพิวเตอร์เป็นงานที่ต้องใช้เวลาในการทำงานค่อนข้างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้โปรแกรม AutoCAD ในการเขียนโปรแกรมค้นหาเส้นทาง

เพราะว่าภาษาทางปัญญาประดิษฐ์ในโปรแกรม AutoCAD นั้นมีอยู่เพียงภาษาเดียวคือ AutoLISP ซึ่งก็คือภาษา LISP นั่นเองแต่ทว่าการทำงานนั้นไม่สนับสนุน (support) ทุกคำสั่งที่มีอยู่ใน LISP จริงๆ ทำให้การเขียนโปรแกรมเป็นไปอย่างยากลำบากและอาจจะใช้งานจริงไม่ได้เลย แต่ทว่าก็ยังมีโปรแกรมอีกพวกหนึ่งที่อาจช่วยได้คือ ADS ซึ่งเป็นการทำงานโดยการเขียนโปรแกรมภาษา C แล้วค่อยเขียนโปรแกรมภาษา LISP ในการโหลดเรียกใช้โปรแกรมนั้นอีกทีหนึ่ง วิธีนี้เป็นวิธีที่อาจเป็นทางเลือกทางสุดท้ายแต่ทว่าการทำงานอาจใช้เวลานานกว่าและการเขียนโปรแกรมก็เป็นไปอย่างยากลำบากกว่า

3. การเรียกใช้โปรแกรมภายนอกเข้ามาสนับสนุน

การเขียนโปรแกรมจากภายนอกเข้ามาสนับสนุนนั้น สามารถเรียกให้ทำงานได้ทันทีถ้าโปรแกรมนั้นมีขนาดใหญ่ไม่เกิน 50 kB แต่ถ้ามีขนาดใหญ่เกินกว่านี้เครื่องอาจจะ hang ได้ทำให้การเขียนโปรแกรมต้องเป็นไปอย่างระมัดระวังและไม่สามารถเขียนโปรแกรมให้ทำงานหลายๆอย่างพร้อมกันได้

4. การระบุชื่อของโหนดแต่ละโหนด เป็นชื่อที่ไม่เป็นทางการทำให้ไม่สามารถระบุได้แน่นอน ประกอบกับว่าโปรแกรม AutoCAD ไม่สามารถใช้ภาษาไทยได้นอกจากต้องสร้างฟอนต์ขึ้นมาใช้เองทำให้การระบุชื่อไปเป็นอย่างยากลำบาก

5. ตัวโปรแกรมแสดงผลยังมีข้อบกพร่องในโหนดที่ทำการย้อนรอย (Backtracking) ทำให้แสดงเส้นบางเส้นที่ไม่ถูกต้อง แต่ก็ยังสามารถดูผลได้ เมื่อตัดเส้นนั้นออกก็ยังเห็นผลลัพธ์เหมือนเดิม

5.2 ข้อเสนอแนะ

1. ควรจะมีการศึกษาวิธีการหาค่าน้ำหนักที่มีความน่าเชื่อถือมากกว่านี้
2. วิธีการในการค้นหาเส้นทางที่มีประสิทธิภาพมากกว่านี้เพื่อให้มีการค้นหาเส้นทางแบบดีที่สุด (Best-First Search), การค้นหาเส้นทางแบบ A* เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าเป็นไปได้ควรรอ Version ของ AutoCAD ให้มีประสิทธิภาพทางด้าน AutoLISP มากกว่านี้หรือไม่ก็หาโปรแกรมทาง CAD/CAM อื่นเข้ามาเพื่อให้การทำงานได้สะดวกรวดเร็วยิ่งขึ้น
4. ควรสามารถใช้ภาษาไทยได้ เพราะว่า AutoCAD ไม่มีฟอนต์ภาษาไทยในตัวเอง ดังนั้นต้องสร้างฟอนต์ขึ้นมาใหม่เอง
5. เนื่องจากโปรแกรมนี้จำเป็นต้องใช้พื้นที่ในหน่วยความจำมาก ดังนั้นถ้าต้องการใช้งานได้ทั้งแผนที่ก็ควรหาเครื่องที่มีหน่วยความจำจำนวนมากมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การใช้คำสั่ง COMPILE ในการสร้าง Shape/Font

คำสั่งนี้จะเป็นคำสั่งในการสร้างตัวอักษร (Shape/Font) ที่จะใช้ Drawing ซึ่งปกติโปรแกรม AutoCAD จะใช้ได้เฉพาะรูปแบบ *.SHX เท่านั้น เมื่อเราสร้างตัวอักษร (Font/Shape) ขึ้นมาใหม่ เครื่องจะยังไม่สามารถเข้าใจได้ ตราบใดที่ยังไม่มีการคอมไพล์

หลังจากที่เราใช้คำสั่ง

Command : COMPILE

โปรแกรม AutoCAD จะแสดงสแตนดาร์ดไฟล์ในไดอะล็อกบ็อกซ์ให้ เราเลือกโดยจะมีข้อความบอกว่า"Select Shape or Font File" และให้เราป้อนชื่อไฟล์ โดยในไดอะล็อกบ็อกซ์จะแสดงรูปแบบ*.shp;pfb ซึ่ง *.shp หมายถึง Shape file และ *.pfb จะหมายถึง Post Script file เมื่อเราป้อนชื่อไฟล์หรือเลือก pick ไฟล์ที่เราต้องการใช้งานแล้วการคอมไพล์ก็จะเริ่มขึ้น ถ้ามีความผิดพลาด (error)) เกิดขึ้น โปรแกรมจะแจ้งถึงชนิดของความผิดพลาดและบรรทัดที่อยู่ในไฟล์ Shape /Font ที่ผิดพลาดเพื่อสะดวกในการแก้ไข ถ้าไม่มีความผิดพลาดเกิดขึ้นโปรแกรมจะแจ้งว่า

COMPILATION SUCCESSFULL Output file name .shx Contain nnn bytes

ก็สามารถเรียกใช้งานได้ทันที

ในไฟล์ *.shp จะเป็นซอร์ส (source) ที่ใช้สร้างฟอนต์ซึ่งจะเป็นเท็กซ์ ไฟล์ที่เราสามารถแก้ไขเพิ่มเติมได้ แต่เมื่อเราผ่านการคอมไพล์แล้วจะกลายเป็น Binary Form ซึ่งยากแก่การแก้ไขเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้กริดช่วยในการดราฟต์

ในการเขียนแบบโดยการใช้คอมพิวเตอร์ช่วยในการดราฟต์ (CAD) จะมีเครื่องมือที่ช่วยในการเขียนแบบอยู่หลายลักษณะ และกริดเป็นคำสั่งหนึ่งของ AutoCAD ที่ใช้เป็นเครื่องมือในการช่วยอ้างอิงให้เกิดความสะดวกและแม่นยำในการ pick จุดหรือหาตำแหน่งของชิ้นงาน โดยจะแสดงกริดในลักษณะเป็นจุดที่ช่วงระยะห่างระหว่างจุดในแกนใดๆ เป็นระยะห่างที่เท่ากัน

```
Command : GRID
Grid spacing (x) or ON/OFF/Snap/Aspect< 0.004> : 0.4
```

ถ้าเราป้อนค่า Grid spacing (x) ลงไป จะเกิดจุดเล็กๆ เป็นลักษณะเฟรม (Frame)ขึ้นในจอภาพ แต่ถึงอย่างไรก็ตามกริดที่ปรากฏขึ้นนี้ จะไม่มีผลต่อรูปภาพเลย จะใช้เป็นเพียงจุดอ้างอิงเท่านั้น แต่ถ้าในกรณีที่ช่องห่างของกริดเมื่อเทียบกับระยะช่วงการแสดงทั้งหมดของสกรีนมีขนาดเล็กมากโปรแกรมก็จะแจ้งว่า

```
Grid too dense to Display
```

เราจึงจำเป็นต้องเปลี่ยนช่วงห่างของการแสดงกริดในจอให้เหมาะสมเสียใหม่ ที่สำคัญอีกประการหนึ่งก็คือกริดจะแสดงเฉพาะในช่วง Limits ที่เราตั้งไว้เท่านั้น ถ้าเราต้องการให้กริดแสดงตรงบริเวณใดเราจะต้องตั้ง Limits ให้สัมพันธ์กับขนาดที่เราต้องการแสดงกริดด้วยและในตัวเลือกอื่น

GRID - ON

จะเป็นการแสดงกริดในจอภาพตามระยะ Grid spacing ที่แสดงไว้

```
Command : GRID
Grid spacin (x<) or ON / OFF/ Snap / Aspect <0.1> : ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GRID - OFF

จะเป็นการไม่ให้มีการแสดงกริดในหน้าจอ

```
Command : GRID
Grid spacing (x) or ON / OFF / Snap / Aspect < 0.1 > : OFF
```

Grid - Aspect

จะเป็นการให้ระยะช่วงห่างระหว่าง Grid ในแนวนอนและแนวตั้ง (แกน Xและแกน Y) มีระยะไม่เท่ากัน ตามค่าที่เราป้อนเข้าไป ถ้าเราใช้คำสั่งนี้

```
Command ; GRID
Grid spacing (x) or ON /OFF / Snap / Aspect < 0.00> : A
Horizontal spacing (x) < 0.5 > : 1
Vertical spacing (x) <0.5 > : 0.5
```

Grid - Snap

ในตัวเลือกนี้จะเป็นประโยชน์ต่อเรามากที่สุด ซึ่งจะเป็นการบังคับให้มีการ Snap ตามคำสั่ง Snap โดยอัตโนมัติ โดยจะมีความสัมพันธ์โดยตรงกับคำสั่ง Snap ถ้าสมมติเราให้ Snapมีการเปลี่ยนค่าระยะของการ Snap ในตัวเลือกนี้จะมีการปรับค่า Grid - Snap ไปตามค่าที่เปลี่ยนโดยอัตโนมัติเช่นกัน

```
Command : GRID
Grid spacing (x) or ON /OFF /Snap / Aspect <0.00> : S
```

การใช้คำสั่ง GRID นี้จะสามารถใช้คอนโทรลคีย์ในการ ON/OFF คำสั่ง GRID ได้เช่นกัน โดยการใช้ Ctrl+G หรือคีย์ F7 วิธีการใช้งานก็คล้ายกับคอนโทรลคีย์อื่นๆ คือ ถ้าสมมติค่าเดิมที่ตั้งไว้เป็น Grid - ON ถ้าเรากด F7 จะกลายเป็นในลักษณะตรงกันข้ามทันทีคือ Grid - OFF และเช่นกัน ถ้าเรากด F7 อีกครั้งหนึ่งก็จะกลายเป็น Grid - OFF เช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ Snap ช่วยเพิ่มความแม่นยำ

Snap เป็นคำสั่งหนึ่งของ AutoCAD เปรียบเสมือนตัวเครื่องมือที่ช่วยให้งานเขียนแบบมีความแม่นยำขึ้น เมื่อเราใช้คำสั่ง Snap บนหน้าจอ ถ้าเราขยับ Crosshairs จะพบว่า Crosshairs จะมีการกระโดดอย่างลงตัว ตามระยะที่เราตั้งไว้เสมอ ทั้งในแนวนอนและแนวตั้ง และจะสามารถใช้ร่วมกับคำสั่ง GRID ได้อย่างมีประสิทธิภาพ ลองศึกษาตัวอย่างนี้ดู

การใช้คำสั่ง Grid ร่วมกับ Snap

(1).	Command : NEW	<เปิด Drawing ใหม่>
(2).	Command : ZOOM	<ขยายช่วงการมองตาม Limits ที่ตั้งไว้>
	ALL/Center/Dynamic/Extents/Left/Previous/Vmax/Window/Scale<(x/xp)> : a	
(3).	Command : SNAP	<สร้างการ Snap>
	Snap spacing or ON/OFF/Aspect/Rotate/Style <0.000> : 0.5	
(4).	Command : GRID	
	Grid spacing(x) or ON/OFF/Snap/Aspect <0.000> : S	
(5).	Command : LINE	<สร้างสามเหลี่ยม>
	From point : 3,8	<ป้อนค่าจุดแรก>
	to point : 3,5	<ป้อนค่าจุดที่สอง>
	to point : 6,5	<ป้อนค่าจุดที่สาม>
	to point : C	<ป้อนรูปภาพ>
(6).	Command : LINE	<สร้างสี่เหลี่ยม>
	From point :	<pick จุดแรก>
	to point :	<pick จุดที่สอง>
	to point :	<pick จุดที่สาม>
	to point :	<pick จุดที่สี่>
	to point :	<pick จุดที่ห้า>
	to point : C	<ปิดสี่เหลี่ยม>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะสังเกตได้ว่าเมื่อเราเลื่อน Crosshairs ไปจุดใดๆ บนหน้าจอจะมีการกระโดดเข้าหาจุดกริดเสมออย่างแม่นยำ ในลักษณะนี้จะช่วยให้เรา pick ตำแหน่งต่างๆได้แม่นยำขึ้น และจากตัวอย่างที่ผ่านมา เราจะได้ค่าของกริดขึ้นตรงกับคำสั่ง Snap ถ้าสมมติเรามีการเปลี่ยนค่า Snap เช่นจาก 0.5 เป็น 0.2 การแสดงกริดบนหน้าจอก็จะเปลี่ยนไปตามระยะ Snap โดยจะมีระยะ 0.2 หน่วยเช่นกัน

Snap - ON

ในตัวเลือกนี้จะมีการยอมให้มีการ Snap เกิดขึ้นในหน้าจอ

```
Command : SNAP
Snap spacing or ON/OFF/Aspect/Rotate/Style<0.2> : ON
```

Snap - OFF

ในตัวเลือกนี้จะเป็นการปิดหรือไม่ให้เกิดการ Snap ขึ้น

```
Command : SNAP
Snap spacing or ON/OFF/Aspect/Rotate/Style<0.2> : OFF
```

Snap - Aspect

ในลักษณะนี้เราจะสามารถกำหนดระยะการ Snap ในแกนนอนและแกนตั้ง แต่ก็มีการข้อจำกัดอยู่ที่ว่าถ้าเราใช้ Snap ในโหมด Style - Isometric ในค่าระยะการ Snap ของในแนวนอนและตั้ง จะไม่สามารถให้ค่าระยะที่แตกต่างกันของแนวนอนและแนวตั้งได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Command : SNAP
Snap spacing or ON/OFF/Aspect/Rotate/Style<0.2> : a
Horizontal spacing<0.25> : 1          <ป้อนค่าระยะ Snap ในแนวนอน>
Vertical spacing<0.25> : 0.5          <ป้อนค่าระยะ Snap ในแนวตั้ง>
SNAP - Rotate
```

ในคำสั่งนี้จะใช้ในกรณีที่รูปของเราไม่ได้อยู่ในขนานในแนวตั้งหรือแนวนอนดังรูปที่4.19 เราจำต้องให้มีการหมุน Snap ไปตามลักษณะรูปที่เราต้องการจะเขียนรูป ในคำสั่ง Snap - Style จะมีตัวเลือกให้คือ

ถามจุด Base point ที่ใช้เป็นจุดในการหมุนและค่า Default ที่โปรแกรมได้ตั้งมาให้แล้วคือจุด (0,0) หรือเราสามารถเลือก pick จุดตำแหน่งที่เป็นจุดหมุนได้

ถามมุมที่ใช้ในการหมุนจากจุดหมุนค่า Default ที่ตั้งไว้คือ 0 องศา และในส่วนนี้จะสามารถป้อนค่าได้ตั้งแต่ - 90 องศา ถึง 90 องศา วัดตามเข็มนาฬิกา

การใช้คำสั่ง SNAP - ROTATE

```
(1).  Command : GRID                          <เป็น Grid ตามการ Snap>
      Grid spacing(x)or ON/OFF/Snap/Aspect<0.000> : S
(2).  Command : SNAP                          <ให้ Snap - Rotate>
      Snap Spacing or ON/OFF/Aspect/Rotate/Style<0.5> : r
      Base point <0,0> :                      <Enter>
      Rotation angle<0> : 30                  <ให้หมุนไป 30 องศา>
```

Snap - Style

ในตัวเลือกนี้จะเป็นการเลือกรูปแบบของการแสดง Snap ในตัวเลือก Standard หรือ Isometric ในสภาวะทั่วไปในการใช้งานปกติจะใช้ค่า Standard แต่ถ้าสมมติเราต้องการเขียนภาพ Isometric ใน Crosshairs ก็จะมีการปรับเปลี่ยนตามมุมที่เราต้องการจะสร้าง รูปจะมีมุมที่ช่วยในการเขียนรูปคือ 30, 90 , 150 , 210 , 270 และ 330 จาก 0 องศา งัดจากแกนนอนไปในทิศทวนเข็มนาฬิกา (ค่าบวก)

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในค่าโหมด Isometric นี้เราจะไม่สามารถตั้งค่าการ Snap ของแนวนอนและแนวตั้งให้มีค่าต่างกันได้เลย

Command : SNAP	<ให้อยู่ในโหมด Iso>
Snap spacing or ON/OFF/Aspect/Rotate/Style<0.2> : S	
Standard/Isometric<s> : I	<เลือก Isometric>
Vertical spacing <0.2> : 1.5	<ป้อนค่าระยะ snap>

เมื่อเราป้อนค่าดังกล่าวเราจะได้ Crosshairs หมุนไปในรูปแบบ <Isoplane Top> แต่ถ้าเราต้องการที่หมุนไปในลักษณะอื่นเช่น <Isoplane Left> หรือ <Isoplane Right> เราจะได้คอนโทรลคีย์ในการควบคุมโดยการกด <Ctrl+E> ในการควบคุมการแสดง Isometric และการใช้ <Ctrl+E> นี้จะเป็นการวนลูปการแสดง Isometric และการใช้ <Ctrl+E> นี้จะเป็นการวนลูปการแสดงรูปแบบต่างๆ ของ Isometric จะสลับกันไปเรื่อยๆ ตราบที่เรากดคอนโทรลคีย์

วิธีการใช้คำสั่ง SAVE, SAVEAS, QSAVE กับการเก็บข้อมูล

ในการเก็บข้อมูลในดาต้าไฟล์ของการเขียนรูป (Draft) เป็นกระบวนการหนึ่งที่สำคัญมาก และในส่วนนี้ AutoCAD ก็มีคำสั่งเกี่ยวกับการจัดเก็บข้อมูลในหลายลักษณะ จุดประสงค์เพื่อสร้างความปลอดภัยของข้อมูลให้มีมากที่สุด และในการใช้คำสั่ง SAVE, SAVEAS, QSAVE จากคำสั่งเหล่านี้ถ้าเราใช้คำสั่งใดก็ตามโปรแกรมจะทำการสร้างไฟล์สำรอง เมื่อเรามีการอัปเดท (update) ข้อมูลทุกครั้งเสมอ โดยจะนำไฟล์ข้อมูลเก่าก่อนที่เราจะอัปเดทไปสร้างเป็นไฟล์สำรอง โดยตั้งเป็นชื่อเดิมแต่เปลี่ยนนามสกุลเป็น .BAK เสมอ

และในกรณีถ้าเกิดมีความผิดพลาดในการอัปเดทข้อมูลขึ้น เราจะสามารถนำไฟล์ Drawing.BAK มาเปลี่ยนชื่อ (Rename) กลับเป็น Drawing.DWG เพื่อนำกลับมาใช้ใหม่ได้อีกเช่นกัน แต่ก็มีข้อยกเว้นถ้าในกรณีที่ป้อนชื่อไฟล์ที่ต้องการจัดเก็บข้อมูลไปเป็นชื่ออื่น ที่ไม่มีชื่อซ้ำอยู่ในไดเรกเทอรี่ย่อย ก็จะไม่มีการสร้าง .BAK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ SAVEAS, SAVE

การใช้คำสั่งนี้โปรแกรมจะมีการถามชื่อไฟล์ที่ต้องการจัดเก็บข้อมูลโดยจะเกิดสับไดอะล็อกซ์บ็อกซ์ให้เราเลือก pick ข้อมูลหรือใช้ค่าตาม Default file เดิม

วิธีการใช้ SAVEAS, SAVE

Command : _SAVEAS	<เลือก Pick Pull-down Menu #1 (File)>
Save Drawing as...	<pick หรือป้อนชื่อไฟล์ลงไป>

การใช้คำสั่ง SAVEAS นี้จะเป็นการจัดเก็บลงในไฟล์ใหม่ ตามที่เราป้อนชื่อใหม่ลงไป แต่คำสั่ง SAVE จะเป็นการจัดเก็บลงในไฟล์ที่มีการ Default ชื่อเดิมไว้อยู่ก่อนแล้ว

การใช้ QSAVE

การใช้คำสั่งนี้จะเป็นการจัดเก็บไฟล์ข้อมูลที่จะไม่มีการถามชื่อหรือป้อนชื่อลงไปเลย จะจัดเก็บลงในชื่อเดิมทันที แต่ถ้าในกรณีที่เรายังมิได้ตั้งชื่อไฟล์ข้อมูลไว้ เช่น การใช้คำสั่ง NEW เมื่อเราใช้คำสั่ง QSAVE โปรแกรมจะมีการให้ป้อนชื่อ หรือ pick ชื่อลักษณะเช่นเดียวกับ SAVEAS

Command : SAVE	<จะไม่มีการถามชื่อหรือให้ป้อนชื่อ>

ที่น่าสนใจอีกประการหนึ่งก็คือ ในค่าระบบตัวแปรชื่อ SAVETIME ซึ่งตัวแปรนี้จะทำหน้าที่จัดเก็บไฟล์ Drawing แบบอัตโนมัติ ตามเวลาที่ได้ตั้งไว้เป็นนาที ซึ่งตัวแปร SAVETIME นี้จะเป็นตัวแปรในแบบ Integer ถ้าสมมติเราตั้งค่าไว้ที่ 0 ก็จะหมายถึงไม่มีการจัดเก็บโดยอัตโนมัติ

Command : SETVAR	<เรียกค่าระบบตัวแปร>
Variable name or ? : SAVETIME	<เรียกตัวแปร>
New value for SAVETIME <120> : 30	<ตั้งไว้ที่ 30 นาที>
ซึ่งจะเซฟไว้ที่ไฟล์ AUTO.SV$ เสมอ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้คำสั่ง LIMITS ในการกำหนดขอบเขต

เราจะใช้คำสั่ง LIMITS ในการกำหนดขอบเขตในการอ้างอิงของ Drawing ไม่ให้รูปมีขนาดเกินจากที่เราทำขอบเขตไว้ โดยจะมีการจำกัดระยะในแนวเฉพาะในแกน X และ Y เท่านั้น (เฉพาะ 2D) เราจะไม่สามารถจำกัดระยะในแกน Z ได้

ปกติเมื่อเราทำการสร้าง Drawing ใหม่ การตั้งค่า Limits ของ Drawing ตัวใหม่นี้จะเป็นไปตาม Prototype ถ้าสมมติเราต้องการเปลี่ยนค่า Limits

เราจะสามารถใช้คำสั่ง LIMITS ในการกำหนดขอบเขตใหม่ได้เช่นกัน และผลของ Limits จะมีผลต่อโปรแกรมอยู่ 3 ประการ

1. เมื่อเราให้ค่าตัวแปร LIMCHECK มีค่าเป็น ON หรือเท่ากับ 1 จะสามารถสร้างรูปได้เฉพาะขอบเขต limits ที่ตั้งไว้เท่านั้น จะไม่สามารถสร้างรูปได้นอกเหนือจากบริเวณส่วนอื่นๆได้
2. การใช้คำสั่ง GRID จะแสดงผลเฉพาะในช่วงการแสดงที่เป็น limits เท่านั้น
3. ในคำสั่ง ZOOM ALL จะแสดงเฉพาะในช่วง limit ที่ตั้งไว้เท่านั้น ยกเว้นในกรณีขนาดของรูปที่เกิดขอบเขต limit ออกไปแล้วจะแสดงแบบ Extend แทน

การใช้ GRID กับ Limits

(1). Command : NEW <สร้างไฟล์ Drawing ใหม่>
Loading acad.lsp... loaded <สร้างเมนูและ Protytype ใหม่>
AutoCAD Release 12 Menu Utilities loaded

(2). Command ; GRID
Grid spacing (x) or ON/OFF/Snap/Aspect <0.000> : 0.3 <สร้าง GRID>

(3). Command : ZOOM <Zoom - 100%>
All/Center/Dynamic/Extents/Left/Previous/Window/Scale(x) : a <Zoom แค่ 75%>
Regenerating drawing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LIMITS - POINT

ปกติการใช้คำสั่ง LIMIT นี้จะเริ่มต้นด้วยโปรแกรมจะถามถึงจุดต่ำสุดทางซ้ายมือซึ่งค่า default โปรแกรมได้ตั้งไว้ที่ <0.000,0.000> ให้เราป้อนค่าหรือเลือก pick จุด และจุดต่อไปโปรแกรมจะถามในลักษณะทแยงมุมคือจุดบนสุดที่อยู่ทางขวามือ ค่า default ที่โปรแกรมตั้งไว้จะเป็นตามPhototype และให้เราป้อนค่าหรือ pick จุดเช่นกัน

```
Command : LIMITS      < เรียกคำสั่ง LIMITS.
Reset Model space limits :
ON / OFF / <Lower left corner > < 0.000,0.000> :   <ใส่ค่าจุดล่างซ้ายสุด>
Upper right corner < 12.000, 9.000 > :  120,90  <ใส่ค่าจุดบนขวาสุด>
```

LIMITS - ON

จะเป็นการยอมให้มีการตรวจสอบค่าขอบเขต limits ถ้าเราใช้คำสั่งนี้จะเป็นการไม่ให้มีการสร้างรูปเกินจากค่า limits ที่ได้ตั้งไว้ จะสร้างรูปได้เฉพาะในขอบเขต limits เท่านั้น

```
Command : LIMITS          < เรียกคำสั่ง LIMITS >
Reset Model space limits :  < ยอมให้มีการตรวจสอบ limits >
ON /OFF / <Lower left corner > < 0.000,0.000>  : ON
```

LIMITS - OFF

จะเป็นการไม่ยอมให้มีการตรวจสอบค่าขอบเขต limits ถ้าเราใช้คำสั่งนี้จะยอมให้มีการสร้างรูปเกินจากค่า limits ที่ตั้งไว้

```
Command :  LIMITS                  <  เรียกคำสั่ง LIMITS >
Reset Model space limits :      <ไม่ยอมให้มีการตรวจสอบ limits>
ON /OFF /  < Lower left corner >   <0.000,0.000>  :  OFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ตัวอักษรหรือฟอนต์ภาษาไทยในโปรแกรม AutoCAD

การใช้ฟอนต์ภาษาไทยของโปรแกรม AutoCAD จะต่างไปจากโปรแกรมอื่นๆ ทั่วไป ซึ่งจำเป็นต้องใช้ฟอนต์ที่สร้างและคอมไพล์ (compile) จากโปรแกรมเท่านั้น โดยโปรแกรมจะมองตัวอักษรในแต่ละตัวในลักษณะรูปร่างๆหนึ่ง และนำมาเรียงต่อๆ กันไปให้เป็นคำขึ้นมา การสร้างฟอนต์หรือลักษณะรูปร่างต่างๆ นี้จะต้องถูกสร้างขึ้นแล้วถูกเก็บในไฟล์ที่มีนามสกุล . SHP แล้วจะต้องการคอมไพล์ก่อนให้มีนามสกุล . SHX

หลักในการสร้างฟอนต์ภาษาไทยนี้ จะอาศัยการตรวจสอบรหัส ASCII จากการกดแป้นพิมพ์เช่น อักษรตัว " ก " ถ้าเราดูที่แป้นพิมพ์ 2 ภาษาจะตรงกับอักษร "D" ในภาษาอังกฤษ ถ้าสมมติเรากำลังใช้ฟอนต์ภาษาไทยอยู่ เมื่อเรากดแป้นอักษร D ก็จะหมายถึงส่งรหัส ASCII เท่ากับ 100 แลเมื่อโปรแกรมตรวจรหัส ASCII เท่ากับ 100 ก็นำอักษรรูปร่าง "ก" มาปะลงบนหน้าจอทันที

ด้วยหลักการนี้จึงสรุปได้ว่า เราจะไม่สามารถใช้อักษรภาษาไทยปนกับอักษรอังกฤษได้ในเวลาเดียวกันเลย (ถ้าจำเป็นต้องใช้ร่วมกัน จะต้องพัฒนาโปรแกรมขึ้นมาใช้เอง (ADS) หรือใช้ฟอนต์พิเศษบน Microsoft Windows 3.1 Thai Edition) และอักษรภาษาไทยนี้ก็ยังสามารถที่จะใช้เครื่องหมายอักษรพิเศษได้เช่นกัน

ตัวอย่างต่อไนี้จะเป็นการสร้างตัวอักษรภาษาไทยจากฟอนต์ต่างๆ เฉพาะบางฟอนต์เท่านั้น

ตัวอักษรภาษาไทยแบบที่ 1
ตัวอักษรภาษาไทยแบบที่ 2
ตัวอักษรภาษาไทยแบบที่ 3
ตัวอักษรภาษาไทยแบบที่ 4
ตัวอักษรภาษาไทยแบบที่ 5
ตัวอักษรภาษาไทยแบบที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ฟอนต์ภาษาไทย

(1). Command : STYLE <เปลี่ยนรูปแบบตัวอักษร>
Text style name (or ?) < STANDARD> : THAI < ตั้งชื่อรูปแบบอักษร>
New style <pick ที่ THAI.SHX>
(2). Command : DTEXT <สร้าง TEXT>
Justify / Style / <Start point > : <pick ตำแหน่ง>
Hight <0.2000> : 0.4 <ให้ความสูง 0.4 หน่วย>
Rotation angle <0> : <Enter>
Text :% u9y ; vyd Ki4KKumpc [[muj@ %%u <พิมพ์ข้อความ>
Text : <Enter >
(3). Command ; STYLE <เรียกคำสั่ง style>
Text style name (or?) <THAI> : THAI - P < ตั้งชื่อรูปแบบตัวอักษร>
New style <pick ที่ THAI - O. SHX>
<Enter จนออกคำสั่ง>
(4). Command : DTEXT <สร้าง TEXT>
Justify / Style / <Start point > : <pick ตำแหน่ง>
<แล้วพิมพ์ข้อความเดียวกันและในทำนองเดียวกันทดลองเปลี่ยนฟอนต์ไปเป็น THAI-N.SHX , THAI2.SHX. THAI89 .SHX โดยใช้ข้อความเดียวกัน>

การใช้ Tablet Mode ในการลอกแบบหรือแผนที่

ข้อได้เปรียบอย่างหนึ่งของดิจิไตเซอร์ก็คือ การที่มีความสัมพันธ์ของตำแหน่งต่างๆ ในทุกๆ จุดของกระดานดิจิไตเซอร์ที่สามารถอ้างอิงต่อกันได้ ซึ่งในส่วนนี้เราจะสามารถนำมาใช้ในการลอกรูปต่างๆ นำเข้าสู่จอภาพตามสเกลที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนและวิธีการก็อบปี้รูปหรือแผนที่เข้าโปรแกรม AutoCAD

1. นำแผนที่ที่ต้องการจะทำลอกมาทำการตีเส้นกริด (grid) ให้เป็นตาราง และมีระยะห่างของเส้นกริดตามขนาดของความละเอียดที่ต้องการ (ยิ่งตีเส้นกริดให้ถี่ก็จะยิ่งมีความถูกต้องสูงตามไปด้วย) โดยจะขึ้นอยู่กับขนาดของข้อมูล และความซับซ้อนของรูป
2. สร้างเส้นกริดบนจอภาพตามขนาดจริงในแผนที่ โดยมีขนาดสเกลตามระยะกริดของภาพกับระยะกริดในหน้าจอให้ตรงกันในลักษณะ 1:1
3. นำรูปแผนที่ที่ต้องการจะลอกลงในจอภาพ มาวางบนดิจิไตเซอร์พร้อมกับคลิป (clip) ให้ติดกับดิจิไตเซอร์อย่าให้มีการเลื่อนได้ (ถ้าเลื่อนก็ต้องเริ่มทำตั้งแต่ข้อ 3 ใหม่)
4. ใช้คำสั่ง TABLET ในตัวเลือก CAL เมื่อโปรแกรม ถามว่า
" Digitize point # 1 " ก็ให้ pick ที่ดิจิไตเซอร์ในจุดล่างซ้ายในแผนที่ เสร็จแล้วโปรแกรมจะถามต่อว่า " Enter coordinates for point # 1 " เราก็เลื่อนดิจิไตเซอร์มายังบริเวณ screen area ในจอภาพ พร้อมทั้ง pick ไปยังจุดล่างซ้ายของจุดบนจอภาพ พยายามให้ 1 ช่องของแผนที่มีขนาดเท่ากับ 1 ช่องบน screen ในหน้าจอให้จำง่ายๆก็คือ 1 ช่อง
กริด = 1 ช่องสกรีน (pick # 1 ไป #A และ pick # 2 ไป #B)
5. ในทำนองเดียวกันโปรแกรมจะถามจุดที่ 2 เราก็จะ pick ไปยังจุดบนขวาบนแผนที่ในช่องเดิม และ จุดบนขวาของจอภาพในช่องเดิมตามลำดับ เสร็จแล้วเราจะได้แท็บเลตที่พร้อมจะใช้งานในการลอกแผนที่ตามต้องการ(ใช้คำสั่ง SKETCH)
6. ในส่วนอื่นๆของแผนที่ (ช่องกริดอื่นๆ ที่เกินจากแท็บเลต) ก็ทำในลักษณะเดียวกัน จนเสร็จสมบูรณ์ทั้งแผ่น

หมายเหตุ

ปกติเราจะใช้คำสั่ง sketch ในการลอกเส้นตามแผนที่ โดยการ pick จุดแล้วลากเส้นและ pick อีกครั้งหนึ่งเพื่อยกปากกาขึ้น และจะกด Enter เพื่อบันทึกลงหน้าจอ เราควรศึกษาคำสั่ง SKETCH อย่างละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการก๊อบปี้รูปเข้าสู่โปรแกรมมีดังนี้

Commamd ; ZOOM <ซูมเต็มลิมิต>

ALL / Center /Dynamic / Extent / Left / Previous / Vmax / Window / < Scale (X/XP)> :a

Command ; TABLET <.ใช้ Tablet>

Option (ON / OFF / CAL / CFG) : CAL <เรียก calibrate>

Digitize point # 1: <pick ตรงกริดล่างซ้าย #1 บนแผนที่>

Enter coordinates for point #1 <pick ตรงกริดล่างซ้าย #A ในจอ>

Digitize point #2 <pick ตรงกริดบนขวา #2 บนแผนที่>

Enter coordinates for point #2 <pick ตรงกริดบนขวา #B ในจอ>

Digitize point #3(or RETURN to end) : <กด Enter เพื่อออกจากโปรแกรม>

Command : SKETCH <.ใช้คำสั่งSKETCH>

Rocord increment <0.1000>: 0.15 <ป้อน 0. 15>

Sketch.PenExitQuitRecordEraseConnect.

<Pendown>.....

<pick จุดแรกแล้วกากบาท + ไปตามเส้นขอบรูป>

.....<Pen up> ... <pick อีกครั้งเพื่อยกปากกาขึ้น>

การใช้แท็บเลตในการ Control Command

ในโปรแกรม AutoCAD รีลีส 12 หลังจากที่เราทำการติดตั้งโปรแกรมเสร็จแล้วในสับไดเรกทอรีของ AutoCAD จะมีสับไดเรกทอรีซ้อนอยู่มีชื่อว่า \ SAMPLE และภายในจะมีไฟล์ Drawing ชื่อ TABLET.DWG ซึ่งเป็น Standard Tablet Menu จะมีขนาดที่ใช้สำหรับ 11" * 11" ของดิจิไตเซอร์ เมื่อเปิดไฟล์ Drawing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเราจะสามารถพล็อต Tablet Menu ออกมา ให้เพื่อใช้เป็นเท็มเพลต Drawing ในดิจิไตเซอร์ของเราได้

ซึ่งใน TABLET .DWG นี้เราจะสามารถใช้คำสั่งต่างๆที่ปรากฏในตารางเท็มเพลตได้ซึ่งผ่านทางดิจิไตเซอร์ โดยการ pick ในตำแหน่งรูปไอคอนมี่เราต้องการใช้คำสั่งนั้นๆ ออกมา โดยมิต้องป้อนคำสั่งใน Command line เลย

ในคำสั่งที่ปรากฏอยู่ในเท็มเพลตต่างๆ นั้น เราจะสามารถ customizing หรือสามารถดัดแปลงเพิ่มเติมคำสั่งเหล่านั้นเพื่อให้เข้ากับงานของเราได้เป็นอย่างดี

วิธีการใช้คำสั่ง Tablet จะเริ่มจากการป้อนคำสั่ง

Command : TABLET

เมื่อเราใช้คำสั่งนี้โปรแกรมจะมีออปชั่นให้เลือกดังนี้

Option (ON/OFF/CAL/CFG) :

ดังมีรายละเอียดในตัวเลือกดังนี้

Tablet-CAL => Calibration

AutoCAD จะใช้เท็บเลตในการเปลี่ยนค่าโคออร์ดิเนตจากจุดต่างๆของแผนที่ที่เราวางบนดิจิไตเซอร์ในการลอกแผนที่เข้าสู่โปรแกรม AutoCAD เพื่อที่ต้องการจะเก็บลักษณะต่างๆของแผนที่เป็น Contour, แม่น้ำ หรือ แม้แต่รูปภาพกราฟิกสวยๆ ลงสู่ไฟล์ Drawing แต่อย่างไรก็ตาม ถ้าเราต้องการกลับไปใช้ดิจิไตเซอร์ในโหมดเดิมที่ใช้เป็นเมาส์ก็ใช้วิธีกดคีย์ F10 และถ้าต้องการลอกแผนที่อีกก็กดคีย์ F10 สลับไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ Tablet Calibrate

Command : TABLET	
Option (ON/OFF/CAL/CFG) : CAL	<ป้อน CAL>
Digitize point #1 :	<pick จุดตำแหน่งล่างซ้าย บนดิจิไตซ์>
Enter coordinate for point #1 :	<ป้อนค่า X,Y ที่จุดล่างซ้าย>
Digitize point #2 :	<pick จุดตำแหน่งบนขวาของจุดเดิม>
Enter coordinate for point #2 :	<ป้อนค่า X,Y ที่จุดบนขวาของจุดเดิมในแผนที่>
Digitize point #3 (or RETURN to END)	<Enter>
Select transformation type...	
Orthogonal </Affine/Project/Repeat table> :	

ในตัวเลือกย่อยจะมีดังนี้

Orthogonal จะใช้ในกรณีที่ค่ามีขนาดเดียวกันทั้งแผ่นในตัวเลือกนี้จะสามารถใช้เพียงสองจุดก็สามารถ calibrate ได้อย่างสมบูรณ์

Affine ถ้าต้องการให้มีความสัมพันธ์ในลักษณะ x และ y Scaling ตามรูปบนดิจิไตเซอร์แต่จะต้องใช้สามจุดจึงจะสามารถ calibrate ได้อย่างสมบูรณ์

Projective ถ้าต้องการสร้างภาพ Plane จาก Perspective Projection หรือจาก Space to Plane และในการ Calibrate จะต้องใช้อย่างน้อยสี่จุดขึ้นไป

Tablet-OFF => ออกจากโหมด Tablet

จะเป็นการออกไปสู่ระบบสกรีนปกติเป็นค่า Tablet OFF และหลังจากที่เราใช้คำสั่งนี้แล้ว จะต้องไปเปลี่ยนค่าตัวแปร (Setvar) TABMODE ให้มีค่าเป็น 0 ด้วยเพื่อใช้งานในโหมดปกติต่อไป วิธีการใช้งานในโหมด OFF นี้คือ

Command : TABLET
Option (ON/OFF/CAL/CFG) : OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tablet-ON => เข้าสู่โหมด Tablet

หลังจากที่เราเปลี่ยนโหมด Tablet เป็น OFF แล้วเราต้องการที่จะเข้าไปใช้ในโหมด Tablet อีกเราจะสามารถเปลี่ยนกลับไปสู่โหมด ON ได้เช่นกัน และหลังจากเปลี่ยนเป็นโหมด ON แล้ว เราจะต้องเปลี่ยนค่าตัวแปร TABMODE ให้มีค่าเป็น 1 ด้วยเช่นกันและวิธีการใช้คำสั่งในโหมด ON คือ

```
Command : TABLET
Option (ON/OFF/CAL/CFG) : ON
```

ถ้าสมมติว่า Tablet ยังไม่ได้รับการ Calibrated ในช่วงการใช้งานแล้วเราจะสามารถใช้คีย์พิเศษควบคุมในการ ON/OFF tablet ได้เช่นกัน โดยการใช้คอนโทรลคีย์ คือ Ctrl+T เป็นคีย์ที่ใช้ให้แท็บเลต มีการ ON/OFF วิธีการใช้งานคือ เมื่อเรากด Ctrl+T 1 ครั้ง จะเป็นการเปลี่ยนแปลงค่าในโหมดตรงกันข้ามกับโหมดเดิม เช่น สมมติว่าเดิมมีการตั้งค่าแท็บเลตเป็นโหมด ON เมื่อเรากดคีย์ 1 ครั้ง จะทำการเปลี่ยนค่าให้เป็นโหมด OFF และเช่นกันเมื่อเรากด Ctrl+T ซ้ำอีกครั้งหนึ่ง ก็จะสลับเปลี่ยนเป็นโหมด ON เหมือนเดิม

Tablet - CFG => Configuration

ในตัวเลือกนี้จะเป็นการปรับระบบเมนูต่างๆในระบบเท็มเพลตให้กับโปรแกรม AutoCAD ให้สามารถเรียกเมนูเหล่านี้นำไปใช้ได้ ในขั้นตอนแรกเราจะต้องทำการพล็อตไฟล์ TABLET.DWG ลงบนกระดาษหรือพลาสติกแผ่นใสให้มีขนาดพอเหมาะกับการใช้งานบนดิจิไตเซอร์ เช่นถ้าเรามีดิจิไตแซอร์ขนาด 12*12 นิ้ว ก็สมควรพล็อตขนาดของเท็มเพลตให้มีขนาดโตสุด 11 นิ้ว โดยให้วางลงในตำแหน่งที่ใช้งานบนดิจิไตเซอร์ คือให้ห่างจากขอบประมาณ 1 นิ้ว พร้อมทั้งติดแผ่นเทปใสให้แน่น เพื่อป้องกันการเลื่อนตัวของแผ่นเท็มเพลต

จะเห็นได้ว่ามีการแบ่งเมนูในเท็มเพลตออกเป็น 4 ส่วนใหญ่ๆ โดยจะมีจุด Donut ในเมนูเป็นจุดแบ่งเมนูออกเป็นส่วนๆ ทั้งหมด 4 ส่วน เมื่อสังเกตในส่วนบนสุดของเท็มเพลตนี้จะมีการแบ่งเป็นคอลัมน์ได้ประมาณ 25 คอลัมน์ และเช่นกันในด้านซ้ายมือสุดจะแบ่งเป็นแถว (Rows) โดยทั้งหมดจะเป็นอักษร A-Y และจากจุดแบ่งเหล่านี้ เราจะใช้แบ่งเมนูของแต่ละส่วนให้เป็นส่วนย่อยในลักษณะเป็นตาราง โดยจะมีการนับแถวกับคอลัมน์ตามลักษณะ การแบ่งในแต่ละเมนู ยกตัวอย่างเช่น ในส่วน Area Menu ที่ 1 จะทำการนับตามคอลัมน์ได้ 25 ช่อง และในส่วนแถวก็จะนับได้ 9 ช่อง คือตั้งแต่ A-I เพื่อนำค่าเหล่านี้ป้อนเข้าโปรแกรมในขณะกำลัง Configuration และวิธี pick จุดต่างๆลงใน Configuration เราจะต้องเลือก pick

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามจุด Donut ที่อยู่ริมสุดด้านซ้าย - ขวาของเมนูในแต่ละพื้นที่ยกตัวอย่างเช่น ใน Area 1 จะมีจุด Donut อยู่ 3 จุด

โปรแกรมจะทำการเก็บค่าพารามิเตอร์ต่างๆลงในไฟล์ ACAD.CFG ดังนั้นเพื่อป้องกันความผิดพลาด เราควรทำสำเนาหรือก็อปปี้ไฟล์ ACAD.CFG ไว้เป็นสำเนาไว้ก่อน ถ้าเกิดความผิดพลาดขึ้นเราก็สามารถนำสำเนาไฟล์ ACAD.CFG มาทับไฟล์ที่เปลี่ยนแปลงให้กลับสภาพเดิมได้อีก

วิธี Configuration ใน AutoCAD Tablet Menu

Command : TABLET <เรียก Tablet>
OPTION (ON/OFF/CAL/CFG) : CFG <ป้อน CFG>
Enter the number of tablet menus disired(0-4) <0> : 4 <ป้อนค่า 4>
Digitize the upper left corner of menus area1 : <pick จุด Donut ซ้ายบนของพื้นที่1>
Digitize the lower left corner of menus area1 : <pick จุด Donut ล่างซ้ายของพื้นที่1>
Digitize the lower right corner of menus area1 : <pick จุด Donut ล่างขวาของพื้นที่1>
Enter the number of columns for menu area 1 : 25 <ป้อนค่า 25>
Enter the number of rows for menu area 1 : 9 <ป้อนค่า 9>
Digitize the upper left corner of menus area 2 : <pick จุด Donut ซ้ายบนของพื้นที่2>
Digitize the lower left corner of menus area 2 : <pick จุด Donut ล่างซ้ายของพื้นที่2>
Digitize the lower right corner of menus area 2 : <pick จุด Donut ล่างขวาของพื้นที่2>
Enter the number of columns for menu area 2 : 11 <ป้อนค่า 25>
Enter the number of rows for menu area 2 : 9 <ป้อนค่า 9>
Digitize the upper left corner of menus area3 : <pick จุด Donut ซ้ายบนของพื้นที่3>
Digitize the lower left corner of menus area3 : <pick จุด Donut ล่างซ้ายของพื้นที่3>
Digitize the lower right corner of menus area3 : <pick จุด Donut ล่างขวาของพื้นที่3>
Enter the number of columns for menu area 3 : 9 <ป้อนค่า 25>
Enter the number of rows for menu area 3 : 13 <ป้อนค่า 9>
Digitize the upper left corner of menus area 4 : <pick จุด Donut ซ้ายบนของพื้นที่4>
Digitize the lower left corner of menus area 4 : <pick จุด Donut ล่างซ้ายของพื้นที่4>
Digitize the lower right corner of menus area 4 : <pick จุด Donut ล่างขวาของพื้นที่4>
Enter the number of columns for menu area 4 : 25 <ป้อนค่า 25>
Enter the number of rows for menu area 4 : 7 <ป้อนค่า 9>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Do you want to respecify the screen pointing area <y> Y <ป้อน Y>
Digitize lower left corner of screen pointing area : <pick ตรงซ้ายล่างของพื้นที่ screen>
Digitize upper right corner of screen pointing area : <pick ตรงขวาบนของพื้นที่ screen>

หลักในการใช้เลเยอร์ (Layer) ในการเขียนรูป

1. โปรแกรมจะให้มีการกำหนดเลเยอร์มาตรฐานไว้ชื่อว่า Layer "0" (ศูนย์) และ Layer "0" นี้จะถูกเรียกใช้งานอัตโนมัติทันทีเมื่อเราเริ่มลงมือเขียนรูปในครั้งแรก และนอกจากนี้ใน Layer "0" เราจะไม่สามารถทำการเปลี่ยนชื่อ (Rename) หรือลบ (Delete) ได้เลย เราควรจะใช้ Layer "0" นี้ในการเขียนเส้นร่างให้กับภาพของเราก่อน แล้วจึงทำการปิด (Layer-OFF) ภายหลัง
2. การตั้งชื่อเลเยอร์ควรจะมีมาตรฐานเป็นของเราเอง ควรจะมีการตกลงกันก่อนในกลุ่มของเรา และควรจะมีความสัมพันธ์กับลักษณะงานในแต่ละชั้น เช่นผังไฟฟ้า ชั้นผังห้องน้ำ ชั้นผังท่อน้ำ เป็นต้น ซึ่งจะเป็นการสะดวกต่อการทำงานและแลกเปลี่ยนไฟล์ข้อมูลซึ่งกันและกัน (ควรตั้งไว้ที่ Phototype Drawing)
3. ควรจะมีการรวมกลุ่มตามลักษณะของจุดประสงค์ให้เป็นหมวดหมู่อย่างชัดเจน เช่น เส้น Grid เส้น Contour ตัวอักษร Text Dimension เป็นต้น เพื่อสะดวกในการ Freeze (แช่แข็ง) และเพื่อที่ว่าในการ Regenerate ภาพจะได้เร็วขึ้น

วิธีใช้คำสั่ง Layer นี้จะมีอยู่ด้วยกัน 2 วิธีคือ

1. Command Line
2. Pull-Down หรือ Screen Menu

ว จะมีวิธีการเรียกดังนี้

Command : LAYER
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งพอจะสรุปได้ดังนี้

? จะเป็นการให้โปรแกรมแสดงรายละเอียดของชื่อที่เราตั้งใจไว้แล้วทั้งหมดของเลเยอร์ และค่าที่เราตั้งไว้ของตัวแปรต่างๆบนจอภาพ

Make จะเป็นการสร้างชื่อเลเยอร์ใหม่ จะใช้เลเยอร์ใหม่นั้นมาเป็นเลเยอร์ที่ใช้อยู่ปัจจุบัน (current) ด้วย แต่ถ้าเกิดเราไปตั้งชื่อเลเยอร์ซ้ำกับชื่อเดิมของเรา ก็จะไม่มีผลอะไร และก็จะปรับเลเยอร์ปัจจุบันไปเป็นชื่อนั้นด้วย ถ้าใช้คำสั่งย่อยนี้ โปรแกรมจะมีการถามดังนี้

```
New current layer <Old-name> : New-name
```

Set จะเป็นการตั้งให้เลเยอร์ปัจจุบัน (current layer) เป็นไปตามชื่อที่ป้อนเข้ามาใหม่นั้น แต่มีข้อแม้ว่าจะตั้งชื่อได้เฉพาะชื่อที่เราเคยสร้างไว้แต่เดิมแล้วเท่านั้น

```
New current layer <Current-name> : New-name
```

แต่ถ้าโปรแกรมหาไม่พบก็จะแจ้งว่า

```
Cannot find layer nnnnn.
```

New จะเป็นการตั้งชื่อของเลเยอร์ใหม่เพียงอย่างเดียว และจะไม่นำชื่อที่ตั้งใหม่นั้นมาเป็นเลเยอร์ปัจจุบัน แต่ถ้าหากเราต้องการที่จะนำชื่อใหม่นี้มาใช้งาน ก็ต้องใช้คู่กับคำสั่ง Set

OFF จะเป็นการปิดไม่ให้มีการแสดงบนจอภาพหรือพล็อตออกทางเครื่องพล็อตในการใช้คำสั่งนี้ โปรแกรมจะให้เราป้อนชื่อของชั้นที่ต้องการปิดและเราอาจจะให้กระทำการ OFF หลายๆชั้นพร้อมกันได้ โดยใช้เครื่องหมาย (,) คั่น หรืออาจจะระบุแบบไวด์การ์ด (wild card) ก็ได้ เช่น

```
Layer name (s) to turn Off : A*
```

ON จะเป็นการยกเลิก คำสั่งปิด (OFF) ตามชื่อที่เราป้อนเข้าไป ให้สามารถกลับมาใช้งานได้ตามปกติ แต่ถ้าใช้คำสั่งนี้กับชั้นที่ Freezen (แช่แข็ง) ก็จะต้องการ Thaw เส้นก่อนจึงจะใช้งานได้ การใช้งานเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Layer name(s) to turn ON : A*
```

Color จะเป็นการให้สีแก่ชั้นเลเยอร์ในแต่ละชั้น โดยจะเป็นการแสดงให้ภาพทั้งหมดในชั้นนั้นเปลี่ยนเป็นสีตามที่ได้ตั้งไว้ และการให้สีนี้ เมื่อเราให้สีแก่ชั้นใด ชั้นนั้นจะ ON โดยอัตโนมัติ ดังนั้นถ้าเรายังคงต้องการให้สีแก่ชั้นใดชั้นหนึ่ง แต่ยังที่จะต้องการ OFF อยู่แบบเดิม ก็ให้ใช้เครื่องหมายลบ (-) แทน เช่น

```
Command : LAYER
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock : C
color : -3                          <ป้อนค่าเป็นลบ>
Layer name(s) for color-3(green) <TEST> :
```

และสีที่เราจะสามารถตั้งได้จะมีถึง 255 สี

Ltype จะเป็นการตั้งชนิดของเส้นในเลเยอร์ปัจจุบันนั้นให้เป็นแบบชนิดใด โดยในแต่ละชั้นจะเลือกใช้ชนิดของเส้นได้เพียงชนิดเดียวเท่านั้น แต่ถ้าสมมติว่าเรามีการเปลี่ยนชนิดของเส้นในเลเยอร์ภาพทุกภาพในเลเยอร์นั้นๆ ก็จะปรับเปลี่ยนตามโดยอัตโนมัติ

การใช้งานของ Ltype

```
Command : LAYER
?/Make/Set/New/ON/OFF/Color/Ltype/Freeze/Thaw/LOck/Unlock : L
Line type (or ?) <Continuous> : ?              <พิมพ์ ?>
Line type(s) to list <*> :                     <Enter>
Loaded Linetypes :
        Name                    Description
-------------------------------------------------------------------------
CENTER                          ----------------------------
CONTINUOUS                      Solidline
HIDDEN                          ----------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้าสมมติมีการแสดงชนิดของเส้น (linetype) เพียง CONTINUOUS เพียงเส้นเดียวก็แสดงว่ายังไม่มีการโหลดเอา linetype ชนิดอื่นๆเข้าสู่หน่วยความจำ

จึงควรออกจากคำสั่ง LAYER ก่อนแล้วใช้คำสั่ง LINETYPE ดังนี้

Command : LINETYPE	<เรียกชนิดของเส้น>
?/Create/Load/Set : L	<ต้องการโหลดเข้ามา>
Line type (s) to load : *	<ต้องการโหลดทุกชนิด>
	<เกิดไดอะล็อกซ์บ็อกซ์ให้เลือก>

หรือถ้ามีการโหลดเรียบร้อยแล้ว โปรแกรมก็จะถามว่า

Linetype HIDDEN in already loads. Reload it? <Y> <Enter>

จากนั้นก็กลับไปใช้คำสั่ง Ltype จากที่ผ่านมา อีกครั้งหนึ่งก็จะพบว่ามีชนิดของเส้นเพิ่มขึ้นอีกมาก นอกจากนี้คำสั่ง LINETYPE ยังจะสัมพันธ์กับคำสั่ง LTSCALE ด้วยคือคำสั่ง LTSCALE นี้จะเป็นตัวคูณของขนาด LINETYPE ซึ่งปกติจะตั้งค่าไว้ที่ 1 ในตัวคูณแฟกเตอร์นี้จะเป็นตัวปรับขนาดระยะช่องต่างๆของชนิดลายเส้น ถ้าเราให้ค่ามากระยะช่องห่างของเส้นก็จะห่างตามไปด้วย วิธีใช้งานเป็นดังนี้

Command : LTSCALE	
New scale factor <1.0000> : 5	<ขนาดช่องห่างโตขึ้น 5 เท่า>

Freeze จะเป็นการแช่เย็น ในชั้นที่เราไม่ต้องการ ให้เกิดการแสดงภาพในชั้นนั้นชั่วคราว ส่วนใหญ่เราจะใช้คำสั่งนี้ในกรณีที่ขนาดข้อมูลหรือรายละเอียดของภาพมาก เมื่อเกิดการ Regenerating Drawing ขึ้นจะใช้เวลาในส่วนนี้มากตามไปด้วย แต่ถ้าเราทำการ Freeze ในบางเลเยอร์ที่ไม่ใช้งานเสียบ้างก็จะย่นระยะเวลาขึ้นอีก การใช้งานมีดังนี้

Layer name(s) to Freeze : name-Freeze

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thaw จะเป็นการยกเลิกคำสั่ง Freeze ที่เราทำการแช่เย็นไว้ให้กลับมาใช้งานได้ตามปกติในแบบเดิม การใช้งานคือ

```
Layer name(s) to thaw : name-Thaw
```

Lock จะเป็นคำสั่งที่เพิ่มเข้ามาในรีลิสนี้ จุดประสงค์ก้เพื่อป้องกันความผิดพลาดที่อาจจะเกิดขึ้นอาทิ เช่น การลบ (Erase) โดยการเลือกชิ้นส่วนของภาพ ซึ่งเราอาจจะพลาดไปลบในชิ้นส่วนที่เราไม่ต้องการเกี่ยวข้องเข้าไปด้วย ทำให้ต้องเสียเวลาในการมาเลือกใหม่ และถ้าเราใช้คำสั่งนี้กับชิ้นใดแล้ว เราจะไม่อาจกระทำการลบ COPY, MOVE หรือทำการดัดแปลงใดๆบนชิ้นที่เราทำการล็อคได้เลย

วิธีการใช้งานมีดังนี้

```
Layer name(s) to Lock : Layer-name
```

และถ้าสมมติเราทำการล็อคในภาพของบางชั้นไว้เลย แต่เรายังเอิญไปทำการลบในส่วนนี้เข้าโปรแกรมก็จะแจ้งว่า

```
Select object : 1 found                <เลือก pick รูป>
1 was on a locked layer                <มีการล็อคโดยเลเยอร์>
Select object :                        <ให้ไปเลือกใหม่อีกครั้ง>
```

Unlock เป็นคำสั่งที่ตรงกันข้ามกับคำสั่ง Lock โดยจะเป็นการยกเลิกคำสั่ง Lock ของเลเยอร์นั้นๆ เพื่อที่จะสามารถใช้งานได้ในแบบปกติ วิธีการใช้งานมีดังนี้

```
Layer name(s) to Unlock : Layer-name
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้คำสั่ง REDRAW

REDRAW เป็นคำสั่งที่ใช้เขียนภาพบนจอใหญ่ และจะทำการลบกากบาท (Blip) ที่ทำเป็นเครื่องหมายไว้ในขณะที่เขียนภาพออกหมด เรามักจะใช้คำสั่งนี้ในกรณีที่เราทำการต่างๆกับ Entities ของเรา เช่น MOVE, ERASE, COPY ฯลฯ โดยจะทำให้ภาพของเราขาดความสมบูรณ์ลงไปได้ ซึ่งเราจะใช้คำสั่งนี้ในการวาดภาพของเราใหม่ แต่จะไม่กระทำการคำนวณภาพใหม่ทั้งหมดให้เรา และเราอาจจะหยุดการ REDRAW ได้โดยการกด Ctrl+C

การใช้คำสั่ง REGEN

คำสั่งนี้จะคล้ายกับคำสั่ง REDRAW แต่โปรแกรมจะทำการคำนวณข้อมูลใหม่ทั้งหมดบนจอภาพของเราเพื่อสร้างรูปขึ้นใหม่ จึงใช้เวลามากกว่าการใช้คำสั่ง REDRAW และในบางคำสั่ง เช่น ZOOM-All, ZOOM-Extents หรืออีกหลายคำสั่งก็จะมีการ REGEN (Regenerating Drawing) อย่างอัตโนมัติ ซึ่งจะสัมพันธ์กับคำสั่ง REGENAUTO

Command : REGEN

คำสั่งต่างๆเหล่านี้เป็นแค่คำสั่งที่ใช้ในการ digitize เท่านั้น รายละเอียดนอกจากนี้ขอให้ผู้สนใจอ่านได้จากหนังสือ AutoCAD ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

รายชื่อโหนดและค่าโคออร์ดิเนตของแต่ละโหนดมีดังนี้

Coordinatesของโหนดชื่อ 1	มีค่าโคออร์ดิเนตเป็น 4.5265, 171.7328
Coordinatesของโหนดชื่อ 2	มีค่าโคออร์ดิเนตเป็น 4.5265, 172.9402
Coordinatesของโหนดชื่อ 3	มีค่าโคออร์ดิเนตเป็น 4.3096, 174.3643
Coordinatesของโหนดชื่อ 4	มีค่าโคออร์ดิเนตเป็น 9.0804, 171.5780
Coordinatesของโหนดชื่อ 5	มีค่าโคออร์ดิเนตเป็น 9.2973, 174.2714
Coordinatesของโหนดชื่อ 6	มีค่าโคออร์ดิเนตเป็น 9.0185, 176.5933
Coordinatesของโหนดชื่อ 7	มีค่าโคออร์ดิเนตเป็น 9.9479, 176.8409
Coordinatesของโหนดชื่อ 8	มีค่าโคออร์ดิเนตเป็น 11.1870, 176.6242
Coordinatesของโหนดชื่อ 9	มีค่าโคออร์ดิเนตเป็น 11.2800, 177.2124
Coordinatesของโหนดชื่อ 10	มีค่าโคออร์ดิเนตเป็น 12.1164, 177.0886
Coordinatesของโหนดชื่อ 11	มีค่าโคออร์ดิเนตเป็น 13.9751, 176.8719
Coordinatesของโหนดชื่อ 12	มีค่าโคออร์ดิเนตเป็น 16.3915, 176.5623
Coordinatesของโหนดชื่อ 13	มีค่าโคออร์ดิเนตเป็น 16.5154, 177.9864
Coordinatesของโหนดชื่อ 14	มีค่าโคออร์ดิเนตเป็น 20.7596, 175.9122
Coordinatesของโหนดชื่อ 15	มีค่าโคออร์ดิเนตเป็น 21.0694, 177.9864
Coordinatesของโหนดชื่อ 16	มีค่าโคออร์ดิเนตเป็น 10.9392, 174.2714
Coordinatesของโหนดชื่อ 17	มีค่าโคออร์ดิเนตเป็น 11.7096, 174.1821
Coordinatesของโหนดชื่อ 18	มีค่าโคออร์ดิเนตเป็น 13.6258, 173.9551
Coordinatesของโหนดชื่อ 19	มีค่าโคออร์ดิเนตเป็น 16.1149, 173.6294
Coordinatesของโหนดชื่อ 20	มีค่าโคออร์ดิเนตเป็น 19.9252, 170.7122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 21	มีค่าโคออร์ดิเนตเป็น 15.8758, 171.0980
Coordinatesของโหนดชื่อ 22	มีค่าโคออร์ดิเนตเป็น 10.6215, 171.4461
Coordinatesของโหนดชื่อ 23	มีค่าโคออร์ดิเนตเป็น 19.3735, 166.7076
Coordinatesของโหนดชื่อ 24	มีค่าโคออร์ดิเนตเป็น 22.3045, 166.3532
Coordinatesของโหนดชื่อ 25	มีค่าโคออร์ดิเนตเป็น 22.1932, 165.7094
Coordinatesของโหนดชื่อ 26	มีค่าโคออร์ดิเนตเป็น 22.0898, 165.0736
Coordinatesของโหนดชื่อ 27	มีค่าโคออร์ดิเนตเป็น 19.1153, 165.4550
Coordinatesของโหนดชื่อ 28	มีค่าโคออร์ดิเนตเป็น 15.4776, 165.9606
Coordinatesของโหนดชื่อ 29	มีค่าโคออร์ดิเนตเป็น 14.9389, 166.0360
Coordinatesของโหนดชื่อ 30	มีค่าโคออร์ดิเนตเป็น 13.5161, 166.1153
Coordinatesของโหนดชื่อ 31	มีค่าโคออร์ดิเนตเป็น 13.5501, 166.5741
Coordinatesของโหนดชื่อ 32	มีค่าโคออร์ดิเนตเป็น 11.9857, 166.7667
Coordinatesของโหนดชื่อ 33	มีค่าโคออร์ดิเนตเป็น 11.8383, 166.8234
Coordinatesของโหนดชื่อ 34	มีค่าโคออร์ดิเนตเป็น 10.1265, 167.0216
Coordinatesของโหนดชื่อ 35	มีค่าโคออร์ดิเนตเป็น 14.7674, 165.5891
Coordinatesของโหนดชื่อ 36	มีค่าโคออร์ดิเนตเป็น 14.0104, 165.6317
Coordinatesของโหนดชื่อ 37	มีค่าโคออร์ดิเนตเป็น 9.9906, 166.0580
Coordinatesของโหนดชื่อ 38	มีค่าโคออร์ดิเนตเป็น 10.0120, 166.2178
Coordinatesของโหนดชื่อ 39	มีค่าโคออร์ดิเนตเป็น 9.9373, 165.6424
Coordinatesของโหนดชื่อ 40	มีค่าโคออร์ดิเนตเป็น 12.2617, 165.3973
Coordinatesของโหนดชื่อ 41	มีค่าโคออร์ดิเนตเป็น 12.1445,164.6408
Coordinatesของโหนดชื่อ 42	มีค่าโคออร์ดิเนตเป็น 13.6585,164.5556
Coordinatesของโหนดชื่อ 43	มีค่าโคออร์ดิเนตเป็น 13.7865,165.2695
Coordinatesของโหนดชื่อ 44	มีค่าโคออร์ดิเนตเป็น 9.8307, 164.8965
Coordinatesของโหนดชื่อ 45	มีค่าโคออร์ดิเนตเป็น 9.0310, 165.7276

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 46	มีค่าโคออร์ดิเนตเป็น 7.1438, 165.8981
Coordinatesของโหนดชื่อ 47	มีค่าโคออร์ดิเนตเป็น 5.1925, 166.0686
Coordinatesของโหนดชื่อ 48	มีค่าโคออร์ดิเนตเป็น 4.3182, 1166.1432
Coordinatesของโหนดชื่อ 49	มีค่าโคออร์ดิเนตเป็น 4.2436, 165.3973
Coordinatesของโหนดชื่อ 50	มีค่าโคออร์ดิเนตเป็น 5.1072, 165.3547
Coordinatesของโหนดชื่อ 51	มีค่าโคออร์ดิเนตเป็น 7.0371, 165.1736
Coordinatesของโหนดชื่อ 52	มีค่าโคออร์ดิเนตเป็น 8.9564, 165.0031
Coordinatesของโหนดชื่อ 53	มีค่าโคออร์ดิเนตเป็น 4.3608, 166.1736
Coordinatesของโหนดชื่อ 54	มีค่าโคออร์ดิเนตเป็น 25.0150,166.0112
Coordinatesของโหนดชื่อ 55	มีค่าโคออร์ดิเนตเป็น 26.1877, 173.7461
Coordinatesของโหนดชื่อ 56	มีค่าโคออร์ดิเนตเป็น 27.0790, 173.7461
Coordinatesของโหนดชื่อ 57	มีค่าโคออร์ดิเนตเป็น 24.8132, 164.6937
Coordinatesของโหนดชื่อ 58	มีค่าโคออร์ดิเนตเป็น 26.6555, 174.5399
Coordinatesของโหนดชื่อ 59	มีค่าโคออร์ดิเนตเป็น 27.5080, 177.5274
Coordinatesของโหนดชื่อ 60	มีค่าโคออร์ดิเนตเป็น 27.2239, 174.9044
Coordinatesของโหนดชื่อ 61	มีค่าโคออร์ดิเนตเป็น 26.0320, 164.9743
Coordinatesของโหนดชื่อ 62	มีค่าโคออร์ดิเนตเป็น 24.7519, 164.0449
Coordinatesของโหนดชื่อ 63	มีค่าโคออร์ดิเนตเป็น 23.1436, 164.2199
Coordinatesของโหนดชื่อ 64	มีค่าโคออร์ดิเนตเป็น 20.7463, 164.5637
Coordinatesของโหนดชื่อ 65	มีค่าโคออร์ดิเนตเป็น 20.6766, 162.2998
Coordinatesของโหนดชื่อ 66	มีค่าโคออร์ดิเนตเป็น 23.0641, 162.1953
Coordinatesของโหนดชื่อ 67	มีค่าโคออร์ดิเนตเป็น 12.0365, 164.8249
Coordinatesของโหนดชื่อ 68	มีค่าโคออร์ดิเนตเป็น 15.1192, 165.1917
Coordinatesของโหนดชื่อ 69	มีค่าโคออร์ดิเนตเป็น 18.1643, 160.4634
Coordinatesของโหนดชื่อ 70	มีค่าโคออร์ดิเนตเป็น 20.5937, 160.2911

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 71	มีค่าโคออร์ดิเนตเป็น 21.3376, 160.2353
Coordinatesของโหนดชื่อ 72	มีค่าโคออร์ดิเนตเป็น 23.8380, 160.1284
Coordinatesของโหนดชื่อ 73	มีค่าโคออร์ดิเนตเป็น 32.4253, 174.0194
Coordinatesของโหนดชื่อ 74	มีค่าโคออร์ดิเนตเป็น 32.9102, 177.0318
Coordinatesของโหนดชื่อ 75	มีค่าโคออร์ดิเนตเป็น 34.1388, 176.8723
Coordinatesของโหนดชื่อ 76	มีค่าโคออร์ดิเนตเป็น 37.2980, 176.0591
Coordinatesของโหนดชื่อ 77	มีค่าโคออร์ดิเนตเป็น 37.5374, 177.2550
Coordinatesของโหนดชื่อ 78	มีค่าโคออร์ดิเนตเป็น 36.4045, 172.2801
Coordinatesของโหนดชื่อ 79	มีค่าโคออร์ดิเนตเป็น 34.2185, 172.6150
Coordinatesของโหนดชื่อ 80	มีค่าโคออร์ดิเนตเป็น 32.3038, 173.1890
Coordinatesของโหนดชื่อ 81	มีค่าโคออร์ดิเนตเป็น 35.8903, 169.7014
Coordinatesของโหนดชื่อ 82	มีค่าโคออร์ดิเนตเป็น 38.1631, 169.0024
Coordinatesของโหนดชื่อ 83	มีค่าโคออร์ดิเนตเป็น 38.6957, 168.8497
Coordinatesของโหนดชื่อ 84	มีค่าโคออร์ดิเนตเป็น 38.9088, 168.7965
Coordinatesของโหนดชื่อ 85	มีค่าโคออร์ดิเนตเป็น 39.6545, 168.5658
Coordinatesของโหนดชื่อ 86	มีค่าโคออร์ดิเนตเป็น 40.8264, 174.0486
Coordinatesของโหนดชื่อ 87	มีค่าโคออร์ดิเนตเป็น 41.9095, 167.7141
Coordinatesของโหนดชื่อ 88	มีค่าโคออร์ดิเนตเป็น 39.5442, 174.9771
Coordinatesของโหนดชื่อ 89	มีค่าโคออร์ดิเนตเป็น 40.0552, 177.9338
Coordinatesของโหนดชื่อ 90	มีค่าโคออร์ดิเนตเป็น 35.0255, 177.8800
Coordinatesของโหนดชื่อ 91	มีค่าโคออร์ดิเนตเป็น 40.7276, 178.0951
Coordinatesของโหนดชื่อ 92	มีค่าโคออร์ดิเนตเป็น 41.5076, 178.1757
Coordinatesของโหนดชื่อ 93	มีค่าโคออร์ดิเนตเป็น 44.1726, 172.6566
Coordinatesของโหนดชื่อ 94	มีค่าโคออร์ดิเนตเป็น 37.3560, 165.6663
Coordinatesของโหนดชื่อ 95	มีค่าโคออร์ดิเนตเป็น 34.9274, 166.3510

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 96 มีค่าโคออร์ดิเนตเป็น 36.8184, 163.3875
Coordinatesของโหนดชื่อ 97 มีค่าโคออร์ดิเนตเป็น 42.8992, 169.6866
Coordinatesของโหนดชื่อ 98 มีค่าโคออร์ดิเนตเป็น 43.1773, 169.9089
Coordinatesของโหนดชื่อ 99 มีค่าโคออร์ดิเนตเป็น 44.1599, 169.5569
Coordinatesของโหนดชื่อ 100 มีค่าโคออร์ดิเนตเป็น 44.8458, 169.0011
Coordinatesของโหนดชื่อ 101 มีค่าโคออร์ดิเนตเป็น 44.0486, 166.9447
Coordinatesของโหนดชื่อ 102 มีค่าโคออร์ดิเนตเป็น 38.6723, 164.5547
Coordinatesของโหนดชื่อ 103 มีค่าโคออร์ดิเนตเป็น 38.4869, 163.6840
Coordinatesของโหนดชื่อ 104 มีค่าโคออร์ดิเนตเป็น 40.2110, 162.9614
Coordinatesของโหนดชื่อ 105 มีค่าโคออร์ดิเนตเป็น 36.6070, 159.5753
Coordinatesของโหนดชื่อ 106 มีค่าโคออร์ดิเนตเป็น 33.6172, 161.0555
Coordinatesของโหนดชื่อ 107 มีค่าโคออร์ดิเนตเป็น 33.2332, 159.8768
Coordinatesของโหนดชื่อ 108 มีค่าโคออร์ดิเนตเป็น 37.3751, 159.1093
Coordinatesของโหนดชื่อ 109 มีค่าโคออร์ดิเนตเป็น 37.7591, 160.9184
Coordinatesของโหนดชื่อ 110 มีค่าโคออร์ดิเนตเป็น 39.1031, 159.9042
Coordinatesของโหนดชื่อ 111 มีค่าโคออร์ดิเนตเป็น 44.6988, 161.7133
Coordinatesของโหนดชื่อ 112 มีค่าโคออร์ดิเนตเป็น 43.6839, 159.8494
Coordinatesของโหนดชื่อ 113 มีค่าโคออร์ดิเนตเป็น 30.4077, 161.2949
Coordinatesของโหนดชื่อ 114 มีค่าโคออร์ดิเนตเป็น 28.1320, 161.4510
Coordinatesของโหนดชื่อ 115 มีค่าโคออร์ดิเนตเป็น 27.9312, 160.0686
Coordinatesของโหนดชื่อ 116 มีค่าโคออร์ดิเนตเป็น 28.4220, 160.0240
Coordinatesของโหนดชื่อ 117 มีค่าโคออร์ดิเนตเป็น 30.3442, 159.9559)
Coordinatesของโหนดชื่อ 118 มีค่าโคออร์ดิเนตเป็น 23.6595, 158.7494
Coordinatesของโหนดชื่อ 119 มีค่าโคออร์ดิเนตเป็น 28.3998, 158.5367
Coordinatesของโหนดชื่อ 120 มีค่าโคออร์ดิเนตเป็น 28.4385, 157.0286

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 121	มีค่าโคออร์ดิเนตเป็น 28.1676, 155.7718
Coordinatesของโหนดชื่อ 122	มีค่าโคออร์ดิเนตเป็น 25.4395, 156.6806
Coordinatesของโหนดชื่อ 123	มีค่าโคออร์ดิเนตเป็น 25.0719, 154.9791
Coordinatesของโหนดชื่อ 124	มีค่าโคออร์ดิเนตเป็น 31.8825, 154.7471
Coordinatesของโหนดชื่อ 125	มีค่าโคออร์ดิเนตเป็น 45.8686, 171.6148
Coordinatesของโหนดชื่อ 126	มีค่าโคออร์ดิเนตเป็น 38.3808, 158.3803
Coordinatesของโหนดชื่อ 127	มีค่าโคออร์ดิเนตเป็น 41.3055, 156.1624
Coordinatesของโหนดชื่อ 128	มีค่าโคออร์ดิเนตเป็น 42.4464, 155.3125
Coordinatesของโหนดชื่อ 129	มีค่าโคออร์ดิเนตเป็น 43.6909, 154.3590
Coordinatesของโหนดชื่อ 130	มีค่าโคออร์ดิเนตเป็น 44.7903, 155.8307
Coordinatesของโหนดชื่อ 131	มีค่าโคออร์ดิเนตเป็น 45.7859, 157.2195
Coordinatesของโหนดชื่อ 132	มีค่าโคออร์ดิเนตเป็น 46.4082, 158.1316
Coordinatesของโหนดชื่อ 133	มีค่าโคออร์ดิเนตเป็น 48.0054, 160.2666
Coordinatesของโหนดชื่อ 134	มีค่าโคออร์ดิเนตเป็น 42.2597, 157.4683
Coordinatesของโหนดชื่อ 135	มีค่าโคออร์ดิเนตเป็น 40.4343, 154.8772)
Coordinatesของโหนดชื่อ 136	มีค่าโคออร์ดิเนตเป็น 43.4213, 156.7635
Coordinatesของโหนดชื่อ 137	มีค่าโคออร์ดิเนตเป็น 46.2837, 154.8150
Coordinatesของโหนดชื่อ 138	มีค่าโคออร์ดิเนตเป็น 47.3001, 156.3075
Coordinatesของโหนดชื่อ 139	มีค่าโคออร์ดิเนตเป็น 49.9968, 164.6170
Coordinatesของโหนดชื่อ 140	มีค่าโคออร์ดิเนตเป็น 52.9713, 163.5897
Coordinatesของโหนดชื่อ 141	มีค่าโคออร์ดิเนตเป็น 54.2179, 166.5404
Coordinatesของโหนดชื่อ 142	มีค่าโคออร์ดิเนตเป็น 46.8536, 174.0059
Coordinatesของโหนดชื่อ 143	มีค่าโคออร์ดิเนตเป็น 48.0100, 177.5954
Coordinatesของโหนดชื่อ 144	มีค่าโคออร์ดิเนตเป็น 48.0450, 178.0156
Coordinatesของโหนดชื่อ 145	มีค่าโคออร์ดิเนตเป็น 48.8860, 177.9631

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 146	มีค่าโคออร์ดิเนตเป็น 50.5856, 177.7004
Coordinatesของโหนดชื่อ 147	มีค่าโคออร์ดิเนตเป็น 54.9309, 176.6498
Coordinatesของโหนดชื่อ 148	มีค่าโคออร์ดิเนตเป็น 24.8305, 153.7618
Coordinatesของโหนดชื่อ 149	มีค่าโคออร์ดิเนตเป็น 27.3130, 153.0124
Coordinatesของโหนดชื่อ 150	มีค่าโคออร์ดิเนตเป็น 30.9849, 151.8237
Coordinatesของโหนดชื่อ 151	มีค่าโคออร์ดิเนตเป็น 34.3466, 150.1182
Coordinatesของโหนดชื่อ 152	มีค่าโคออร์ดิเนตเป็น 35.8206, 152.7540
Coordinatesของโหนดชื่อ 153	มีค่าโคออร์ดิเนตเป็น 37.9927, 151.7202
Coordinatesของโหนดชื่อ 154	มีค่าโคออร์ดิเนตเป็น 36.7579, 150.1775
Coordinatesของโหนดชื่อ 155	มีค่าโคออร์ดิเนตเป็น 34.9695, 151.2304
Coordinatesของโหนดชื่อ 156	มีค่าโคออร์ดิเนตเป็น36.0135, 149.2404
Coordinatesของโหนดชื่อ 157	มีค่าโคออร์ดิเนตเป็น 37.0576, 150.2064
Coordinatesของโหนดชื่อ 158	มีค่าโคออร์ดิเนตเป็น 38.1983, 149.7234
Coordinatesของโหนดชื่อ 159	มีค่าโคออร์ดิเนตเป็น 39.2230, 151.1145
Coordinatesของโหนดชื่อ 160	มีค่าโคออร์ดิเนตเป็น 39.1470, 153.1615
Coordinatesของโหนดชื่อ 161	มีค่าโคออร์ดิเนตเป็น 41.5649, 154.0125
Coordinatesของโหนดชื่อ 162	มีค่าโคออร์ดิเนตเป็น 40.2403, 152.3421
Coordinatesของโหนดชื่อ 163	มีค่าโคออร์ดิเนตเป็น 39.4203, 151.2705
Coordinatesของโหนดชื่อ 164	มีค่าโคออร์ดิเนตเป็น 45.1332, 153.2672
Coordinatesของโหนดชื่อ 165	มีค่าโคออร์ดิเนตเป็น 48.5604, 153.1258
Coordinatesของโหนดชื่อ 166	มีค่าโคออร์ดิเนตเป็น 49.8039, 154.7322
Coordinatesของโหนดชื่อ 167	มีค่าโคออร์ดิเนตเป็น 47.3473, 151.4790
Coordinatesของโหนดชื่อ 168	มีค่าโคออร์ดิเนตเป็น 48.5200, 150.6000
Coordinatesของโหนดชื่อ 169	มีค่าโคออร์ดิเนตเป็น 51.1182, 153.9441
Coordinatesของโหนดชื่อ 170	มีค่าโคออร์ดิเนตเป็น 50.7644, 156.4699

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 171	มีค่าโคออร์ดิเนตเป็น 51.8259, 158.2278
Coordinatesของโหนดชื่อ 172	มีค่าโคออร์ดิเนตเป็น 54.1613, 156.8235
Coordinatesของโหนดชื่อ 173	มีค่าโคออร์ดิเนตเป็น 51.3002, 153.8128
Coordinatesของโหนดชื่อ 174	มีค่าโคออร์ดิเนตเป็น 52.7661, 152.7015
Coordinatesของโหนดชื่อ 175	มีค่าโคออร์ดิเนตเป็น 54.3331, 151.6204
Coordinatesของโหนดชื่อ 176	มีค่าโคออร์ดิเนตเป็น 57.1133, 155.0353
Coordinatesของโหนดชื่อ 177	มีค่าโคออร์ดิเนตเป็น 59.6307, 153.4592
Coordinatesของโหนดชื่อ 178	มีค่าโคออร์ดิเนตเป็น 56.9718, 149.9433
Coordinatesของโหนดชื่อ 179	มีค่าโคออร์ดิเนตเป็น 51.5226, 148.3477
Coordinatesของโหนดชื่อ 180	มีค่าโคออร์ดิเนตเป็น 49.9960, 149.5392
Coordinatesของโหนดชื่อ 181	มีค่าโคออร์ดิเนตเป็น 40.6980, 150.4126
Coordinatesของโหนดชื่อ 182	มีค่าโคออร์ดิเนตเป็น 37.3972, 148.5978
Coordinatesของโหนดชื่อ 183	มีค่าโคออร์ดิเนตเป็น 39.6869, 149.0440
Coordinatesของโหนดชื่อ 184	มีค่าโคออร์ดิเนตเป็น 38.8626, 147.9228
Coordinatesของโหนดชื่อ 185	มีค่าโคออร์ดิเนตเป็น 41.6872, 146.6898
Coordinatesของโหนดชื่อ 186	มีค่าโคออร์ดิเนตเป็น 40.8893, 145.6341
Coordinatesของโหนดชื่อ 187	มีค่าโคออร์ดิเนตเป็น 41.8333, 144.8817
Coordinatesของโหนดชื่อ 188	มีค่าโคออร์ดิเนตเป็น 42.7660, 144.1967
Coordinatesของโหนดชื่อ 189	มีค่าโคออร์ดิเนตเป็น 43.7774, 145.4208
Coordinatesของโหนดชื่อ 190	มีค่าโคออร์ดิเนตเป็น 44.3280, 144.9940
Coordinatesของโหนดชื่อ 191	มีค่าโคออร์ดิเนตเป็น 43.6650, 144.0170
Coordinatesของโหนดชื่อ 192	มีค่าโคออร์ดิเนตเป็น 42.7098, 142.7930
Coordinatesของโหนดชื่อ 193	มีค่าโคออร์ดิเนตเป็น 43.4290, 142.1641
Coordinatesของโหนดชื่อ 194	มีค่าโคออร์ดิเนตเป็น 41.9681, 141.8833
Coordinatesของโหนดชื่อ 195	มีค่าโคออร์ดิเนตเป็น 41.3725, 142.3662

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 196	มีค่าโคออร์ดิเนตเป็น 40.4623, 143.0849
Coordinatesของโหนดชื่อ 197	มีค่าโคออร์ดิเนตเป็น 39.5408, 143.8037
Coordinatesของโหนดชื่อ 198	มีค่าโคออร์ดิเนตเป็น 46.1822, 148.0935
Coordinatesของโหนดชื่อ 199	มีค่าโคออร์ดิเนตเป็น 44.0134, 145.2411
Coordinatesของโหนดชื่อ 200	มีค่าโคออร์ดิเนตเป็น 48.8567, 145.8138
Coordinatesของโหนดชื่อ 201	มีค่าโคออร์ดิเนตเป็น 46.6092, 143.2085
Coordinatesของโหนดชื่อ 202	มีค่าโคออร์ดิเนตเป็น 14.5267, 160.5322
Coordinatesของโหนดชื่อ 203	มีค่าโคออร์ดิเนตเป็น 14.1563, 157.7605
Coordinatesของโหนดชื่อ 204	มีค่าโคออร์ดิเนตเป็น 17.7249, 157.4987
Coordinatesของโหนดชื่อ 205	มีค่าโคออร์ดิเนตเป็น 9.1988, 158.4861
Coordinatesของโหนดชื่อ 206	มีค่าโคออร์ดิเนตเป็น 17.3017, 154.9772
Coordinatesของโหนดชื่อ 207	มีค่าโคออร์ดิเนตเป็น 14.0344, 155.4390
Coordinatesของโหนดชื่อ 208	มีค่าโคออร์ดิเนตเป็น 11.1136, 155.9831
Coordinatesของโหนดชื่อ 209	มีค่าโคออร์ดิเนตเป็น 8.7209, 156.3624
Coordinatesของโหนดชื่อ 210	มีค่าโคออร์ดิเนตเป็น 9.2655, 160.9637
Coordinatesของโหนดชื่อ 211	มีค่าโคออร์ดิเนตเป็น 13.6384, 152.2069
Coordinatesของโหนดชื่อ 212	มีค่าโคออร์ดิเนตเป็น 10.7836, 152.4872
Coordinatesของโหนดชื่อ 213	มีค่าโคออร์ดิเนตเป็น 16.7901, 151.8441
Coordinatesของโหนดชื่อ 214	มีค่าโคออร์ดิเนตเป็น 8.2589, 152.7511
Coordinatesของโหนดชื่อ 215	มีค่าโคออร์ดิเนตเป็น 8.0444, 152.5801
Coordinatesของโหนดชื่อ 216	มีค่าโคออร์ดิเนตเป็น 4.1006, 153.0973
Coordinatesของโหนดชื่อ 217	มีค่าโคออร์ดิเนตเป็น 3.9850, 161.1776
Coordinatesของโหนดชื่อ 218	มีค่าโคออร์ดิเนตเป็น 7.5824, 147.8040
Coordinatesของโหนดชื่อ 219	มีค่าโคออร์ดิเนตเป็น 6.1467, 148.0513
Coordinatesของโหนดชื่อ 220	มีค่าโคออร์ดิเนตเป็น 6.4768, 151.7781

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 221	มีค่าโคออร์ดิเนตเป็น 4.2656, 152.0585
Coordinatesของโหนดชื่อ 222	มีค่าโคออร์ดิเนตเป็น 4.7936, 149.1067
Coordinatesของโหนดชื่อ 223	มีค่าโคออร์ดิเนตเป็น 4.7111, 148.1668
Coordinatesของโหนดชื่อ 224	มีค่าโคออร์ดิเนตเป็น 4.0180, 147.2268
Coordinatesของโหนดชื่อ 225	มีค่าโคออร์ดิเนตเป็น 5.6517, 148.0678
Coordinatesของโหนดชื่อ 226	มีค่าโคออร์ดิเนตเป็น 5.0576, 143.9782
Coordinatesของโหนดชื่อ 227	มีค่าโคออร์ดิเนตเป็น 7.0370, 143.6979
Coordinatesของโหนดชื่อ 228	มีค่าโคออร์ดิเนตเป็น 5.0576, 143.9782
Coordinatesของโหนดชื่อ 229	มีค่าโคออร์ดิเนตเป็น 5.1896, 143.9453
Coordinatesของโหนดชื่อ 230	มีค่าโคออร์ดิเนตเป็น 4.4141, 141.7850
Coordinatesของโหนดชื่อ 231	มีค่าโคออร์ดิเนตเป็น 4.7276, 142.6425
Coordinatesของโหนดชื่อ 232	มีค่าโคออร์ดิเนตเป็น 5.4207, 142.4446
Coordinatesของโหนดชื่อ 233	มีค่าโคออร์ดิเนตเป็น 5.1731, 141.6861
Coordinatesของโหนดชื่อ 234	มีค่าโคออร์ดิเนตเป็น 6.8563, 142.1313
Coordinatesของโหนดชื่อ 235	มีค่าโคออร์ดิเนตเป็น 6.8068, 141.6696
Coordinatesของโหนดชื่อ 236	มีค่าโคออร์ดิเนตเป็น 14.2904, 141.7933
Coordinatesของโหนดชื่อ 237	มีค่าโคออร์ดิเนตเป็น 22.1733, 150.8525
Coordinatesของโหนดชื่อ 238	มีค่าโคออร์ดิเนตเป็น 23.1094, 155.4806
Coordinatesของโหนดชื่อ 239	มีค่าโคออร์ดิเนตเป็น 29.5143, 149.2776
Coordinatesของโหนดชื่อ 240	มีค่าโคออร์ดิเนตเป็น 29.7606, 148.7847
Coordinatesของโหนดชื่อ 241	มีค่าโคออร์ดิเนตเป็น 24.1933, 141.7933
Coordinatesของโหนดชื่อ 242	มีค่าโคออร์ดิเนตเป็น 31.8299, 147.5045
Coordinatesของโหนดชื่อ 243	มีค่าโคออร์ดิเนตเป็น 35.8206, 144.4520
Coordinatesของโหนดชื่อ 244	มีค่าโคออร์ดิเนตเป็น 34.2933, 142.0395
Coordinatesของโหนดชื่อ 245	มีค่าโคออร์ดิเนตเป็น 38.5796, 142.2364

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 246	มีค่าโคออร์ดิเนตเป็น 51.6137, 143.4846
Coordinatesของโหนดชื่อ 247	มีค่าโคออร์ดิเนตเป็น 54.2753, 146.3893
Coordinatesของโหนดชื่อ 248	มีค่าโคออร์ดิเนตเป็น 49.8277, 141.4198)
Coordinatesของโหนดชื่อ 249	มีค่าโคออร์ดิเนตเป็น 58.5127, 143.4496)
Coordinatesของโหนดชื่อ 250	มีค่าโคออร์ดิเนตเป็น 60.1936, 147.6142)
Coordinatesของโหนดชื่อ 251	มีค่าโคออร์ดิเนตเป็น 56.7617, 149.7139
Coordinatesของโหนดชื่อ 252	มีค่าโคออร์ดิเนตเป็น 62.5049, 151.3588
Coordinatesของโหนดชื่อ 253	มีค่าโคออร์ดิเนตเป็น 65.1314, 149.6439
Coordinatesของโหนดชื่อ 254	มีค่าโคออร์ดิเนตเป็น 62.8902, 146.0743
Coordinatesของโหนดชื่อ 255	มีค่าโคออร์ดิเนตเป็น 61.5594, 141.2798
Coordinatesของโหนดชื่อ 256	มีค่าโคออร์ดิเนตเป็น 57.8123, 141.1748
Coordinatesของโหนดชื่อ 257	มีค่าโคออร์ดิเนตเป็น 63.1353, 146.4593
Coordinatesของโหนดชื่อ 258	มีค่าโคออร์ดิเนตเป็น 63.7306, 142.2597
Coordinatesของโหนดชื่อ 259	มีค่าโคออร์ดิเนตเป็น 64.0108, 140.5799
Coordinatesของโหนดชื่อ 260	มีค่าโคออร์ดิเนตเป็น 59.5282, 153.2486
Coordinatesของโหนดชื่อ 261	มีค่าโคออร์ดิเนตเป็น 64.4661, 153.5635
Coordinatesของโหนดชื่อ 262	มีค่าโคออร์ดิเนตเป็น 67.1976, 152.7236
Coordinatesของโหนดชื่อ 263	มีค่าโคออร์ดิเนตเป็น 65.3065, 149.8189
Coordinatesของโหนดชื่อ 264	มีค่าโคออร์ดิเนตเป็น 67.4077, 148.7340
Coordinatesของโหนดชื่อ 265	มีค่าโคออร์ดิเนตเป็น 68.2832, 148.9440
Coordinatesของโหนดชื่อ 266	มีค่าโคออร์ดิเนตเป็น 68.1782, 150.0639
Coordinatesของโหนดชื่อ 267	มีค่าโคออร์ดิเนตเป็น 66.2521, 151.2888
Coordinatesของโหนดชื่อ 268	มีค่าโคออร์ดิเนตเป็น 69.4111, 149.4029
Coordinatesของโหนดชื่อ 269	มีค่าโคออร์ดิเนตเป็น 68.5284, 146.4943
Coordinatesของโหนดชื่อ 270	มีค่าโคออร์ดิเนตเป็น 72.4156, 146.5993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 271 มีค่าโคออร์ดิเนตเป็น 73.0809, 142.7147
Coordinatesของโหนดชื่อ 272 มีค่าโคออร์ดิเนตเป็น 68.9836, 142.6097
Coordinatesของโหนดชื่อ 273 มีค่าโคออร์ดิเนตเป็น 72.8008, 151.0788
Coordinatesของโหนดชื่อ 274 มีค่าโคออร์ดิเนตเป็น 70.1743, 157.2381
Coordinatesของโหนดชื่อ 275 มีค่าโคออร์ดิเนตเป็น 68.7385, 155.1734
Coordinatesของโหนดชื่อ 276 มีค่าโคออร์ดิเนตเป็น 63.2754, 157.8681
Coordinatesของโหนดชื่อ 277 มีค่าโคออร์ดิเนตเป็น 64.5011, 159.3729
Coordinatesของโหนดชื่อ 278 มีค่าโคออร์ดิเนตเป็น 69.7058, 152.2541
Coordinatesของโหนดชื่อ 279 มีค่าโคออร์ดิเนตเป็น 67.0844, 152.5680
Coordinatesของโหนดชื่อ 280 มีค่าโคออร์ดิเนตเป็น 72.6896, 151.3366
Coordinatesของโหนดชื่อ 281 มีค่าโคออร์ดิเนตเป็น 69.3593, 140.3231
Coordinatesของโหนดชื่อ 282 มีค่าโคออร์ดิเนตเป็น 69.0060, 142.4764
Coordinatesของโหนดชื่อ 283 มีค่าโคออร์ดิเนตเป็น 76.6180, 142.9947
Coordinatesของโหนดชื่อ 284 มีค่าโคออร์ดิเนตเป็น 74.8319, 146.7393
Coordinatesของโหนดชื่อ 285 มีค่าโคออร์ดิเนตเป็น 74.0965, 148.2091
Coordinatesของโหนดชื่อ 286 มีค่าโคออร์ดิเนตเป็น 70.8747, 148.9090
Coordinatesของโหนดชื่อ 287 มีค่าโคออร์ดิเนตเป็น 69.6276, 149.5060)
Coordinatesของโหนดชื่อ 288 มีค่าโคออร์ดิเนตเป็น 73.5537, 140.3294
Coordinatesของโหนดชื่อ 289 มีค่าโคออร์ดิเนตเป็น 73.3582, 150.7856
Coordinatesของโหนดชื่อ 290 มีค่าโคออร์ดิเนตเป็น 74.8455, 153.6857
Coordinatesของโหนดชื่อ 291 มีค่าโคออร์ดิเนตเป็น 75.3222, 156.2233
Coordinatesของโหนดชื่อ 292 มีค่าโคออร์ดิเนตเป็น 49.0117, 145.6972
Coordinatesของโหนดชื่อ 293 มีค่าโคออร์ดิเนตเป็น 76.3349, 149.6307
Coordinatesของโหนดชื่อ 294 มีค่าโคออร์ดิเนตเป็น 71.3971, 159.1147
Coordinatesของโหนดชื่อ 295 มีค่าโคออร์ดิเนตเป็น 76.2298, 157.6099

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 296	มีค่าโคออร์ดิเนตเป็น 75.9147, 156.1050
Coordinatesของโหนดชื่อ 297	มีค่าโคออร์ดิเนตเป็น 72.9029, 160.8645
Coordinatesของโหนดชื่อ 298	มีค่าโคออร์ดิเนตเป็น 75.6345, 162.5543
Coordinatesของโหนดชื่อ 299	มีค่าโคออร์ดิเนตเป็น 74.5839, 164.0142
Coordinatesของโหนดชื่อ 300	มีค่าโคออร์ดิเนตเป็น 73.5101, 164.3806
Coordinatesของโหนดชื่อ 301	มีค่าโคออร์ดิเนตเป็น 72.1710, 161.8230
Coordinatesของโหนดชื่อ 302	มีค่าโคออร์ดิเนตเป็น 71.5447, 162.6432
Coordinatesของโหนดชื่อ 303	มีค่าโคออร์ดิเนตเป็น 71.2531, 163.3230
Coordinatesของโหนดชื่อ 304	มีค่าโคออร์ดิเนตเป็น 72.6928, 166.2890
Coordinatesของโหนดชื่อ 305	มีค่าโคออร์ดิเนตเป็น 70.6266, 167.2339
Coordinatesของโหนดชื่อ 306	มีค่าโคออร์ดิเนตเป็น 69.9262, 165.7640
Coordinatesของโหนดชื่อ 307	มีค่าโคออร์ดิเนตเป็น 69.2609, 164.2592
Coordinatesของโหนดชื่อ 308	มีค่าโคออร์ดิเนตเป็น 67.2297, 165.3091
Coordinatesของโหนดชื่อ 309	มีค่าโคออร์ดิเนตเป็น 65.8639, 161.6344
Coordinatesของโหนดชื่อ 310	มีค่าโคออร์ดิเนตเป็น 64.2997, 159.4134
Coordinatesของโหนดชื่อ 311	มีค่าโคออร์ดิเนตเป็น 61.2413, 160.5146
Coordinatesของโหนดชื่อ 312	มีค่าโคออร์ดิเนตเป็น 62.3296, 163.6292
Coordinatesของโหนดชื่อ 313	มีค่าโคออร์ดิเนตเป็น 63.4826, 167.1989
Coordinatesของโหนดชื่อ 314	มีค่าโคออร์ดิเนตเป็น 64.0779, 168.6337
Coordinatesของโหนดชื่อ 315	มีค่าโคออร์ดิเนตเป็น 67.8601, 166.9539
Coordinatesของโหนดชื่อ 316	มีค่าโคออร์ดิเนตเป็น 68.0299, 166.6839
Coordinatesของโหนดชื่อ 317	มีค่าโคออร์ดิเนตเป็น 68.3153, 168.3887
Coordinatesของโหนดชื่อ 318	มีค่าโคออร์ดิเนตเป็น 64.7281, 169.7886
Coordinatesของโหนดชื่อ 319	มีค่าโคออร์ดิเนตเป็น 64.5332, 170.0686
Coordinatesของโหนดชื่อ 320	มีค่าโคออร์ดิเนตเป็น 62.6071, 170.6985

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 321	มีค่าโคออร์ดิเนตเป็น 63.4125, 173.0082
Coordinatesของโหนดชื่อ 322	มีค่าโคออร์ดิเนตเป็น 61.6616, 173.2882
Coordinatesของโหนดชื่อ 323	มีค่าโคออร์ดิเนตเป็น 60.5409, 175.6680
Coordinatesของโหนดชื่อ 324	มีค่าโคออร์ดิเนตเป็น 60.5409, 174.0931
Coordinatesของโหนดชื่อ 325	มีค่าโคออร์ดิเนตเป็น 59.1051, 176.1929
Coordinatesของโหนดชื่อ 326	มีค่าโคออร์ดิเนตเป็น 58.6849, 169.3686
Coordinatesของโหนดชื่อ 327	มีค่าโคออร์ดิเนตเป็น 59.5604, 109.1586
Coordinatesของโหนดชื่อ 328	มีค่าโคออร์ดิเนตเป็น 61.6616, 176.8228
Coordinatesของโหนดชื่อ 329	มีค่าโคออร์ดิเนตเป็น 63.0623, 176.7878
Coordinatesของโหนดชื่อ 330	มีค่าโคออร์ดิเนตเป็น 64.2530, 176.4379
Coordinatesของโหนดชื่อ 331	มีค่าโคออร์ดิเนตเป็น 64.1480, 174.5481
Coordinatesของโหนดชื่อ 332	มีค่าโคออร์ดิเนตเป็น 65.6538, 173.9531
Coordinatesของโหนดชื่อ 333	มีค่าโคออร์ดิเนตเป็น 64.1480, 169.9286
Coordinatesของโหนดชื่อ 334	มีค่าโคออร์ดิเนตเป็น 69.9613, 172.4483
Coordinatesของโหนดชื่อ 335	มีค่าโคออร์ดิเนตเป็น 69.0858, 170.2785
Coordinatesของโหนดชื่อ 336	มีค่าโคออร์ดิเนตเป็น 73.6384, 168.3187
Coordinatesของโหนดชื่อ 337	มีค่าโคออร์ดิเนตเป็น 75.2843, 167.5488
Coordinatesของโหนดชื่อ 338	มีค่าโคออร์ดิเนตเป็น 73.6985, 164.3140
Coordinatesของโหนดชื่อ 339	มีค่าโคออร์ดิเนตเป็น 73.4760, 168.4213
Coordinatesของโหนดชื่อ 340	มีค่าโคออร์ดิเนตเป็น 74.5071, 170.8170
Coordinatesของโหนดชื่อ 341	มีค่าโคออร์ดิเนตเป็น 75.3062, 172.4991
Coordinatesของโหนดชื่อ 342	มีค่าโคออร์ดิเนตเป็น 70.5566, 174.1631
Coordinatesของโหนดชื่อ 343	มีค่าโคออร์ดิเนตเป็น 66.2842, 175.6680
Coordinatesของโหนดชื่อ 344	มีค่าโคออร์ดิเนตเป็น 67.1597, 177.6977
Coordinatesของโหนดชื่อ 345	มีค่าโคออร์ดิเนตเป็น 71.3971, 176.1579

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Coordinatesของโหนดชื่อ 346 มีค่าโคออร์ดิเนตเป็น 74.1637, 175.1430
Coordinatesของโหนดชื่อ 347 มีค่าโคออร์ดิเนตเป็น 75.1092, 178.0127
Coordinatesของโหนดชื่อ 348 มีค่าโคออร์ดิเนตเป็น 76.1598, 174.4781
Coordinatesของโหนดชื่อ 349 มีค่าโคออร์ดิเนตเป็น 75.9497, 170.3485
Coordinatesของโหนดชื่อ 350 มีค่าโคออร์ดิเนตเป็น 63.4476, 177.8027
Coordinatesของโหนดชื่อ 351 มีค่าโคออร์ดิเนตเป็น 59.5604, 177.8727
Coordinatesของโหนดชื่อ 352 มีค่าโคออร์ดิเนตเป็น 57.0894, 161.6694
Coordinatesของโหนดชื่อ 353 มีค่าโคออร์ดิเนตเป็น 55.6731, 158.3448
Coordinatesของโหนดชื่อ 354 มีค่าโคออร์ดิเนตเป็น 52.9700, 163.5880
Coordinatesของโหนดชื่อ 355 มีค่าโคออร์ดิเนตเป็น 58.2646, 162.9293
Coordinatesของโหนดชื่อ 356 มีค่าโคออร์ดิเนตเป็น 58.0195, 164.5741
Coordinatesของโหนดชื่อ 357 มีค่าโคออร์ดิเนตเป็น 58.4747, 165.7990
Coordinatesของโหนดชื่อ 358 มีค่าโคออร์ดิเนตเป็น 54.8192, 167.9930
Coordinatesของโหนดชื่อ 359 มีค่าโคออร์ดิเนตเป็น 51.8210, 169.7536
Coordinatesของโหนดชื่อ 360 มีค่าโคออร์ดิเนตเป็น 55.5681, 178.0127
Coordinatesของโหนดชื่อ 361 มีค่าโคออร์ดิเนตเป็น 48.2298, 150.8231

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ขนาดแผนที่ที่ทั้งหมดและขอบเขตแผนที่ที่ใช้แสดงมีดังต่อไปนี้

รูปที่ ค.1 แสดงแผนที่ที่ ตัดออกมาเพื่อใช้กับโปรแกรมค้นหาเส้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.2 แสดงแผนที่รวมทั้งหมดของถนนเยาวราช

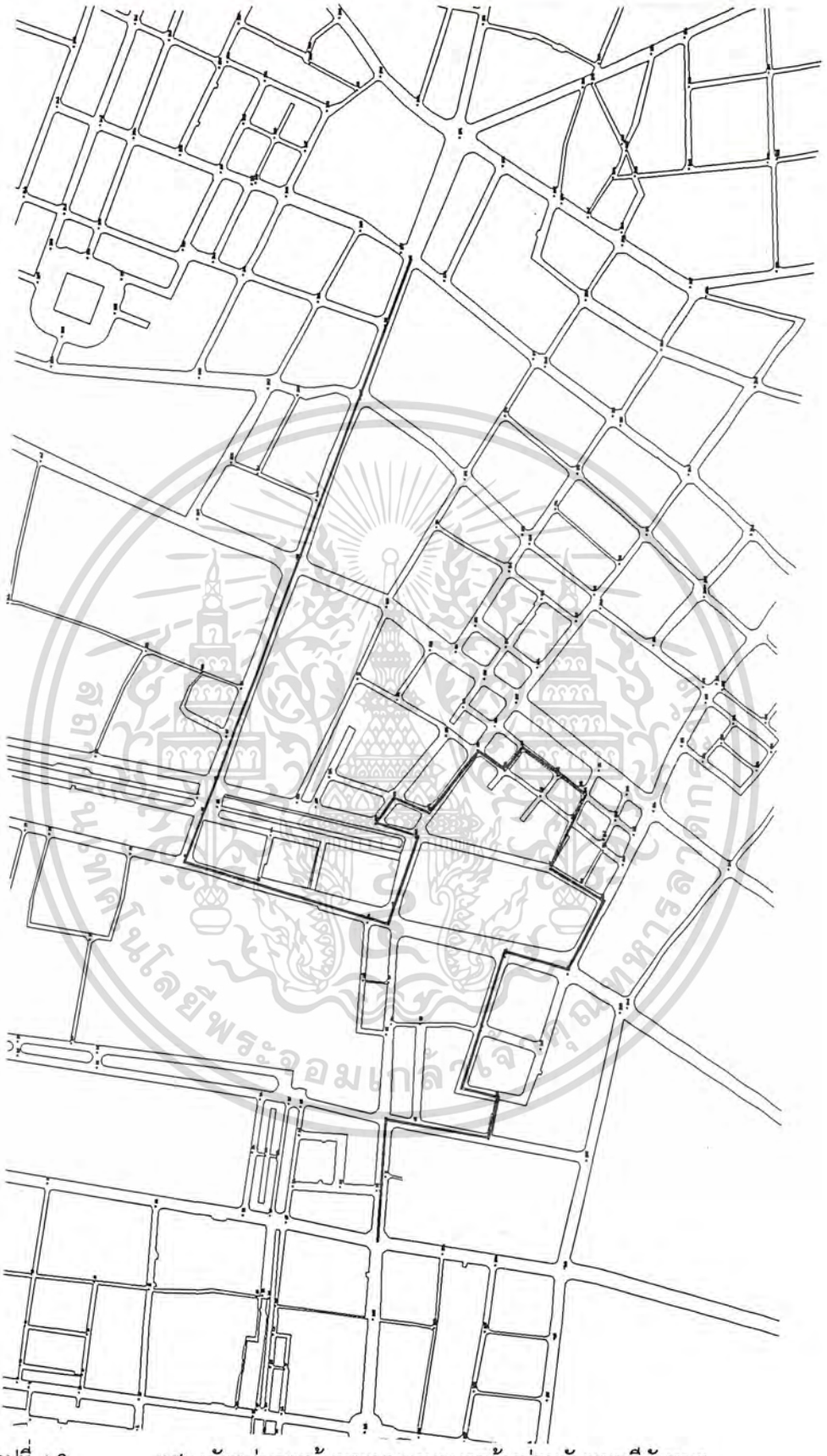
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

ภาพแสดงผลลัพธ์ที่เป็นผลจากการค้นหาเส้นทางในแต่ละวิธีเมื่อใช้กับแผนที่ทั้งหมด

รูปที่ ง.1 แสดงตัวอย่างการค้นหาแบบตามแนวลึกก่อนกับแผนที่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.2 แสดงตัวอย่างการค้นหาแบบตามแนวกว้างก่อนกับแผนที่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ง.3 แสดงตัวอย่างการค้นหาแบบปีนตามเนินเขากับแผนที่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ง.4 แสดงตัวอย่างการค้นหาแบบเสียค่าใช้จ่ายน้อยที่สุดกับแผนที่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ รศ.ดรรชิต ไมตรี อาจารย์ที่ปรึกษาคณะวิศวกรรมศาสตร์ ภาควิชาคอมพิวเตอร์ ที่ให้คำปรึกษา แนะแนวทาง ตลอดจนช่วยแก้ปัญหาในการทำปริญญานิพนธ์นี้จนทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี และขอขอบคุณอาจารย์ทุกท่าน ที่ได้ประสิทธิประสาทวิชาต่างๆ

ขอขอบคุณภาควิชาวิศวกรรมเครื่องกลที่อนุญาตให้ยืมเครื่อง digitizer มาใช้ในการเก็บแผนที่

ขอขอบคุณกองบังคับการตำรวจจราจรที่อนุญาตให้นำแผนที่บริเวณถนนเยาวราชมาใช้กับปริญญานิพนธ์นี้

ขอบคุณพ่อแม่ พี่น้อง ที่เข้าใจ เป็นกำลังใจ และคอยสนับสนุนในการทำงานตลอดเวลา

ขอขอบคุณเพื่อนๆ น้องๆ ทุกคนที่ได้ให้กำลังใจในการทำปริญญานิพนธ์นี้จนสำเร็จลุล่วงไปด้วยดี โดยเฉพาะ นางสาวพรรณสิริ ติยะพรรณ นายบรรจบ ปานปลอด นายอนันต์ ตันฑิกุล นางสาวปิยะฉัตร สงวนทรัพย์ ที่ช่วยให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. อนิรุต ลิ่วหาทอง, "พื้นฐานการโปรแกรม AutoLISP", บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 368 หน้า, 2536
2. กอบเกียรติ สระอุบล, "Advanced AutoCAD", บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 375 หน้า, 2536
3. วิทยา สงวนวรรณ, "เริ่มต้นสมบูรณ์แบบกับ AutoCAD Release 12", บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), หน้า 1-626, 861-1105, 2537
4. ดร. ก่อเกียรติ เก่งสกุล และ บุญเจริญ ศิริเนาวกุล, "ทฤษฎีและการประยุกต์ใช้งานปัญญาประดิษฐ์และระบบผู้เชี่ยวชาญ", บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), หน้า 27-60
5. David J. Steele, "Golden Common LISP A Hands-on Approach", ADDISON-WESLEY PUBLISHING COMPANY, 543 p., 1991
6. Herbert Schildt, "Advanced Turbo Prolog Version 1.1", BORLAND-OSBORNE/McGRAW-HILL, 323 p., 1987
7. George F.Luger and William A.Stubblefield, "Artificial Intelligence and the Design of Expert Systems", The Benjamin/Cummings Publishing Company Inc., 660 p., 1988
8. Patrick Henry Winston and Berthold Klaus Paul Horn, "LISP", ADDISON-WESLEY PUBLISHING COMPANY, 619 p., 1990
9. John D. Hood, CET, "Using AutoCAD with AutoLISP", McGraw-Hill Publishing Company, 239 p., 1992

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้