สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# โปรแกรมแนะนำเส้นทางของรถเมล์ในกรุงเทพมหานคร

# Bangkok Mass Transportation Bus Navigator

นาย มนูญ ธัญญวิบูลย์
นาย เมธา ตันติวัฒนาศิริกุล

เลขหมู่.....
เลขทะเบียน 42785
วัน, เดือน, ปี 10 ส.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรมแนะนำเส้นทางของสายรถเมล์ในกรุงเทพมหานคร

# Bangkok Mass Transportation Bus Navigator


โดย

นาย มนูญ ธัญญวิบูลย์

นาย เมธา ตันติวัฒนาศิริกุล

อาจารย์ที่ปรึกษา

อ. เกียรติกูล เจียรนัยธนะกิจ


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมแนะนำเส้นทางของสายรถเมล์ในกรุงเทพมหานคร

Bangkok Mass Transportation Bus Navigator

ผู้จัดทำ

1. นาย มนูญ ธัญญวิบูลย์ รหัสประจำตัว 40010586
2. นาย เมธา ตันติวัฒนาศิริกุล รหัสประจำตัว 40010604

อาจารย์ที่ปรึกษา

(อ.เกียรติกูล เจียรนัยธนะกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรมแนะนำเส้นทางของสายรถเมล์ในกรุงเทพมหานคร

นาย มนูญ ธัญญวิบูลย์ 40010586
นาย เมธา ตันติวัฒนาศิริกุล 40010604
อ.เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

## บทคัดย่อ

โปรแกรมแนะนำและค้นหาเส้นทางสายรถเมล์นี้ ทำขึ้นมาเพื่อแนะนำเส้นทางที่ต่อรถน้อยที่สุดให้กับผู้ใช้ในกรุงเทพ ฯ เนื่องด้วยในปัจจุบันยังไม่มีการเว็บที่บอกหรือแนะนำเส้นทางรถเมล์โดยแสดงผลเป็นรูปภาพแผนที่ โดยใช้ภาษาจาวา (Java Language) ซึ่งเป็นภาษาที่ไม่ขึ้นอยู่กับแพลตฟอร์มและมีลักษณะการออกแบบโปรแกรมเป็นแบบอ็อบเจ็กต์-โอเรียนเต็ด (Object-Oriented) ซึ่งสามารถอิมพลีเมนต์ได้ง่าย จะมีใช้อัลกอริทึมในการค้นหาแบบ เบรดท์-เฟิรสท์-เซิรสช์ (Breadth-First Search) ซึ่งจะเป็นการทำโปรแกรมในช่วงแรกคือในการทำงานช่วงเทอมต้น

โดยอัลกอริทึมประเภทนี้จะมีหลักการในการค้นหาเส้นทางไปตามระดับชั้น ( level ) จากนั้นในช่วงหลังของการทำงานจะใช้อัลกอริทึมอีกตัวหนึ่งซึ่งคือ อัลกอริทึมที่ชื่อ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ (Iterative Deepening Search ) เพื่อเป็นการหาทางที่ดีที่สุด และแสดงผลผ่านทางเว็บเพจ โดยใช้จาวา แอ็พเพล็ต (Java Applet) ซึ่งเป็นภาษาที่ความสามารถสูงในการเขียนเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Bangkok Mass Transportation Bus Navigator**

Manoon Tanyaviboon

Metha Tantiwattanasirikul

Kietkul Jearanaitanakij Advisor

## ABSTRACT

Bangkok Mass Transportation Bus Navigator is designed to suggest the minimal taking bus route in Bangkok. Currently, there is no web application that shows or suggests the bus route in map display. Java Language is independent from platform and has Object-Oriented Programming Design which is easy to implement , and can be part of components for coding .Breadth-first search algorithm is implemented in the first term , this algorithm has rule for search in level by level. In the finai term has additional algorithm, 'Iterative Deepening Search',which is the space -optimize search algorithm . Output will be displayed in the web browser. Java Applet that has a high ability for Web Programming, is used for this project as well.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ เกียรติกูล เจียรนัยธนกิจ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

มนูญ ธัญญวิบูลย์

เมธา ตันติวัฒนาศิริกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | หน้าที่ |
|---|---|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญภาพ | VIII |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความสำคัญและที่มา | 1 |
| 1.2 วัตถุประสงค์ของงานวิจัย | 2 |
| 1.3 ขอบเขตของงานวิจัย | 3 |
| 1.4 วิธีการดำเนินงาน | 4 |
| บทที่ 2 ภาษาจาวา | 5 |
| 2.1 ภาษาจาวา | 4 |
| 2.2 จาวาเวอร์ชวลแมชีน (Java's Virtual Machine) | 9 |
| 2.3 จาวาแอพเพล็ต | 10 |
| บทที่ 3 โครงสร้างข้อมูล | 14 |
| 3.1 สแตก (stack) | 14 |
| 3.2 คิว (queue) | 14 |
| 3.3 ทรี (tree) | 15 |
| บทที่ 4 ทฤษฎีและการออกแบบ | 17 |
| 4.1 ยูเอ็มแอล (UML) | 17 |
| 4.1.1 เท็กซ์โน้ต (Text Note) | 18 |
| 4.1.2 คอนสเตรน (Constrain) | 18 |
| 4.1.3 สเตริโอไทป์ (Steleotype) | 18 |
| 4.2 รูปแบบของการออกแบบ ( Design Pattern ) | 19 |
| 4.3 ดีไซน์แพตเทิร์นที่มีการนำมาใช้ในระบบ | 20 |
| 4.4 การนำมาประยุกต์ใช้ในการออกแบบ | 22 |
| 4.5 การออกแบบส่วนการค้นหา | 23 |
| บทที่ 5 อัลกอริทึมการค้นหา | 26 |
| 5.1 อัลกอริทึมการค้นหา | 26 |
| 5.1.1 ประสิทธิภาพของอัลกอริทึม | 26 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 บิ๊ก-โอ โนเตชัน (THE BIG-OH NOTATION) 28

5.2 การแก้ปัญหาโดยใช้การค้นหา 29

5.3 การค้นหาทางที่ใช้สำหรับแก้ปัญหา 31

5.3.1 สร้างลำดับของการกระทำ 31

5.3.2 โครงสร้างข้อมูลสำหรับทรีการค้นหา 33

5.4 การวางแผนการค้นหา 34

5.4.1 เบรดท์-เฟิรสท์ เซิรสช์ (Breadth-first search) 35

5.4.2 เดพท์-เฟิรสท์ เซิรสช์ (Depth-first search) 36

5.4.3 เดพท์-ลิมิต เซิรสช์ (Dep th-limited search) 38

5.4.4 อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ (Iterative deepening search) 38

5.5 การเปรียบเทียบวิธีการค้นหา 40

5.6 ตัวอย่างการค้นหาเบรดท์-เฟิรสท์ เซิรสช์ 41

บทที่ 6 ส่วนติดต่อกับผู้ใช้ 45

6.1 ส่วนการรับอินพุตข้อมูล 45

6.2 ส่วนการแสดงผลข้อมูล 48

6.2.1 ส่วนการแสดงผลบนแผนที่ใหญ่ 48

6.2.2 ส่วนการแสดงผลบนแผนที่ขนาดขยายใหญ่ 49

บทที่ 7 หลักการแสดงผล 50

7.1 ส่วนการแสดงผลบนแผนที่ใหญ่ 51

7.2 ส่วนการแสดงผลบนแผนที่ที่ขยายขนาด 52

7.3 การนำข้อมูลมาแสดงผลทิศทาง 58

7.3.1 แสดงผลทิศทางส่วนที่เป็นจุดสิ้นสุดของแผนที่ 58

7.3.2 แสดงผลทิศทางในช่วงถนน 59

7.3.3 แสดงผลทิศทางช่วง 3 แยกและ 4 แยก 59

7.3.4 แสดงผลทิศทางช่วง 5 แยก 60

7.4 การแสดงสายรถเมล์ 61

7.5 ขอบเขตการทำงาน 62

บทที่ 8 ตัวอย่างการทำงาน 65

8.1 ตัวอย่างการทำงาน 65

8.1 1 การค้นหาเส้นทางโดยใช้อัลกอริทึม เบรดท์-เฟิรสท์ เซิรสช์ 66

8.1.2 การค้นหาเส้นทางโดยใช้อัลกอริทึมอิเทอเรทีฟ ดีพเพ็นดิ่ง เซิรสช์ 68

8.2 ผลการทดสอบวิธีการค้นหาทั้ง 2 วิธี

บทที่ 9 วิจารณ์และสรุป 72

ภาคผนวก 73


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. รายละเอียดรถเมล์แต่ละสายที่ใช้ในโครงงาน 74
ภาคผนวก ข. รายละเอียดโหนดที่แทนแต่ละป้ายในโครงงาน 78
บรรณานุกรม 84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

หน้าที่

ตารางที่ 5-1 การประเมินวิธีการค้นหา 40
ตารางที่ 7-1 แสดงการเก็บค่าข้อมูลของโหนดแต่ละโหนด 50
ตารางที่ 7-2 แสดงหลักการคำนวณวิธีแสดงผลของโหนดที่เป็นแยก 60
ตารางที่ 7-3 แสดงหลักการคำนวณวิธีแสดงผลของโหนดที่เป็น 5 แยก 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ

หน้าที่

รูปที่ 3-1 การทำงานของสแตก 14
รูปที่ 3-2 คิวอย่างง่าย 15
รูปที่ 3-3 ทายาทของโจ 15
รูปที่ 4-1 โครงสร้างของดีไซน์แพตเทิร์น โมเดล-วิว-คอนโทรลเลอร์ 20
รูปที่ 4-2 โครงสร้างการทำงานของการดีไซน์แพตเทิร์น โมเดล-วิว-คอนโทรลเลอร์ 21
รูปที่ 4-3 โครงสร้างของดีไซน์แพตเทิร์นของโปรเจกต์ 22
รูปที่ 4-4 โครงสร้างของดีไซน์แพตเทิร์นของโปรเจกต์ 23
รูปที่ 4-5 แสดงการติดต่อข้อมูลในส่วนของการค้นหาเส้นทาง 23
รูปที่ 4-6 แสดงออบเจ็กต์ และการติดต่อระหว่างกัน 24
รูปที่ 5-1 เอเจนต์แก้ไขปัญหาแบบง่าย 31
รูปที่ 5-2 ทรีการค้นหาบางส่วนสำหรับการหาเส้นทางจากอะราดไปยังบูราเชส 32
รูปที่ 5-3 คำอธิบายอย่างไม่เป็นทางการของอัลกอริทึมการค้นหาแบบธรรมดา 33
รูปที่ 5-4 อัลกอริทึมการค้นหาแบบธรรมดา 34
รูปที่ 5-5 เบรดท์-เฟิรสท์ เซิรสช์ทรีหลังจากการขยายโหนด 0,1,2, และ 3 36
รูปที่ 5-6 เดพท์-เฟิรสท์ เซิรสช์ทรีสำหรับทรีฐาน 2 โหนดที่ความลึก 3 37
รูปที่ 5-7 อัลกอริทึม อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ 39
รูปที่ 5-8 การเดินซ้ำ 4 ครั้งของ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ 39
รูปที่ 6-1 แสดงหน้าจอการอินพุทข้อมูลในส่วนบน 45
รูปที่ 6-2 แสดงตัวอย่างตัวเลขกำกับจุดบนแผนที่ 47
รูปที่ 6-3 แสดงตัวอย่างของการแสดงผลข้อมูลบนแผนที่ขนาดใหญ่ 48
รูปที่ 7-2 แสดงแผนที่ขนาดขยายที่เกิดจากการแบ่งรูปขนาดใหญ่มาเป็นรูปเมื่อทำการขยาย 52
รูปที่ 7-3 แสดงแผนที่เมื่อทำการขยายใหญ่ 53
รูปที่ 7-4 รูปแสดงตัวอย่างป้ายรถเมล์แบบต่าง ๆ ที่ใช้ในแผนที่ 54
รูปที่ 7-5 แสดงขอบเขตการทำงานแสดงโดยแผนที่ใหญ่ 63
รูปที่ 8-1 แผนที่แสดงเส้นทางรถเมล์และป้ายรถเมล์ 65
รูปที่ 8-2 กราฟแสดงการเปรียบเทียบหน่วยความจำที่ใช้ 70
รูปที่ 8-3 กราฟแสดงการใช้เวลาที่ใช้ 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและที่มา

ในหัวข้อของโครงงานชิ้นนี้คือเรื่องโปรแกรมแนะนำเส้นทางของสายรถเมล์ในกรุงเทพมหานคร จะมีที่มาอยู่ 2 เรื่อง คือเรื่องของอัลกอริทึมที่ใช้ในการค้นหา และ เหตุผลในการนำไปใช้ในการใช้ค้นหาข้อมูลรถเมล์ โดยทั้ง 2 เรื่องมีรายละเอียดดังนี้

ตั้งแต่อดีตจนถึงปัจจุบันนั้นเรื่องของการค้นหา มีการพัฒนาอัลกอริทึมเพิ่มเติมตลอดเวลา เพื่อให้การค้นหาข้อมูลมีความรวดเร็วและถูกต้อง รวมถึงมีความเหมาะสมในการแก้ปัญหาต่าง ๆ ที่มีในปัจจุบัน โดยวิธีการค้นหาที่มีใช้กันทั่วไป เช่น เบรดท์เฟิรสท์เซิรสช์ เดปท์เฟิรสท์เซิรสช์ เป็นอัลกอริทึมขั้นพื้นฐานที่มีใช้กันทั่วไปในปัจจุบัน โดยอัลกอริทึมทั้ง 2 แบบนี้ต่าง ก็มีข้อดีและข้อเสียที่ต่างกันไป จึงทำให้มีการค้นหาอัลกอริทึมที่เหมาะสมที่ใช้แทนอัลกอริทึมพื้นฐานทั้ง2 ตัวนี้

โดยเริ่มแรกเราจะมาสนใจในเรื่องของอัลกอริทึมพื้นฐานที่ชื่อว่า เบรดท์เฟิรสท์เซิรสช์ เนื่องจากเราไม่สามารถใช้อัลกอริทึมของ เดปท์เฟิรสท์เซิรสช์ ได้เพราะว่า อัลกอริทึมเดปท์เฟิรสท์เซิรสช์ ไม่สามารถหาทางที่ดีที่สุดได้และถ้าเป็นทรีที่มีไม่ทราบความลึกจะทำให้ไม่สามารถกลับออกมาที่รูทได้ จึงทำให้ไม่สามารถค้นหาได้จบ โดยจะได้อธิบายต่อไปในเรื่องของเนื้อหา และ นอกจากนี้ยังมี การศึกษาในอัลกอรทึม ที่ชื่อว่า อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ ซึ่งเหมือนกับเป็นการนำข้อดีของทั้ง 2 วิธีการค้นหาข้างต้น มาใช้ และลดข้อด้อย บางประการ แต่ก็เช่นกัน คือยังมีจุดด้อยอยู่บางประการ ดังจะได้กล่าวต่อไปในเนื้อหา

ในปัจจุบันมีการนำเอาความสามารถ ทางด้านคอมพิวเตอร์มาช่วยในการดำเนินงานด้านต่าง ๆ ในชีวิตประจำวันมากขึ้น ซึ่งในหัวข้อโครงงานนี้เราจะพูดถึงเรื่องการนำเอาคอมพิวเตอร์มาประยุกต์ใช้กับข้อมูลเส้นทางสายรถเมล์ โดยจะนำมาช่วยในการแนะนำและค้นหาเส้นทางของสายรถเมล์ในกรุงเทพฯ ในขณะนี้มีเพียงหน่วยงานเดียวที่รับผิดชอบในเรื่องนี้ คือหน่วยประชาสัมพันธ์ในองค์การขนส่งมวลชนกรุงเทพ หรือหน่วยงาน 184 ซึ่งยังไม่มีการประยุกต์ใช้คอมพิวเตอร์ในการช่วยเหลือการตอบปัญหาให้กับผู้ใช้ โดยจะใช้เพียงประสบการณ์ของผู้ตอบเท่านั้น และใช้การสื่อสารผ่านทางคำพูด ซึ่งทำให้ ข้อมูลที่ได้รับอาจมีความผิดพลาด และไม่สามารถอธิบายข้อมูลผ่านทางคำพูดให้เป็นที่เข้าใจได้

ด้วยข้อจำกัดเหล่านี้ทำให้ จึงมีความจำเป็นต้องนำคอมพิวเตอร์มาช่วยในการดำเนินงาน โดยสร้างเป็นโปรแกรมแนะนำเส้นทางสายรถเมล์ ซึ่งจะทำงานบนเครือข่ายอินเตอร์เนต ใช้ภาษาจาวาในการเขียนโปรแกรม ซึ่งข้อจำกัดต่าง ๆ ที่กล่าวมาข้างต้นนั้นจะถูกแก้ไขด้วยการนำคอมพิวเตอร์มาช่วย โดยในเรื่องการแสดงผลโปรแกรมจะแสดงผลเป็นในรูปแบบของรูปแผนที่ซึ่งจะเข้าใจได้ง่ายกว่าการสื่อสารด้วยวิธีอื่นโดยแผนที่จะแสดงเส้นทางที่จะไปให้ผู้ใช้เห็น และในส่วนของการการค้นหาจะใช้อัลกอริทึมในการค้นหา ซึ่งจะทำให้ข้อมูลที่ได้มีความเที่ยงตรงมากกว่าการใช้ ประสบการณ์ของผู้ตอบตามการค้นหาปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยที่มาทั้ง 2 ประการคือเพื่อใช้ในการช่วยค้นหาเส้นทางสายรถเมล์ในกรุงเทพฯ และเพื่อศึกษาเกี่ยวกับอัลกอริทึมต่าง ๆ ที่เหมาะสม เพื่อเปรียบเทียบประสิทธิภาพ ตามที่ได้กล่าวมาข้างต้นนั้น จึงเป็นที่มาที่เกิดขึ้นของโครงงานชิ้นนี้

ส่วนในด้านความสำคัญนั้น ด้วยเหตุที่ว่าโปรแกรมนี้จะสามารถช่วยให้การเดินทางโดยรถเมล์ไปในกรุงเทพมหานครมีความสะดวกขึ้น เนื่องจากปัจจุบันมีปัญหาในการเดินทางจำนวนมากเนื่องจากผู้ใช้ขาดความเข้าใจในเส้นทางเดินของสายรถเมล์นั้น ๆ จึงทำให้เกิดการเสียเวลาในการเดินทางโดยไม่จำเป็น โปรแกรมดังกล่าวจึงหวังที่จะช่วยในการค้นหาเส้นทางสายรถเมล์ พร้อมทั้งแสดงผลให้อยู่ในรูปของแผนที่เพื่อให้ผู้ใช้ได้ให้รูปจริงของการเดินทาง เพื่อการเดินทางที่รวดเร็วยิ่งขึ้น ด้วยเหตุผลดังกล่าวจึงมีเหตุผลสำคัญในการพัฒนาโปรแกรม

สรุปจากด้วยที่มาและเหตุผลตามที่ได้กล่าวมาข้างต้นแล้วนั้น จึงเกิดเป็นที่มาของโครงงานโปรแกรมแนะนำเส้นทางสายรถเมล์ในกรุงเทพมหานครนี้ขึ้นมา

**1.2 วัตถุประสงค์ของงานวิจัย**

1.2.1 ศึกษาความเป็นไปได้ในการนำเอาอัลกอริทึมแบบ เบรดท์เฟิรสท์ เซิรสช์ มาใช้ในการค้นหาข้อมูลการต่อรถเมล์ รวมถึงการค้นหาอื่น ๆ ว่ามีความเหมาะสมเพียงใดในการดำเนินงาน รวมถึงข้อดี ข้อด้อยที่เกิดขึ้น จากการศึกษาโดยใช้อัลกอริทึมแบบ เบรดท์เฟิรสท์ เซิรสช์

1.2.2 ศึกษาความเป็นไปได้ในการนำเอาอัลกอริทึมแบบ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ มาใช้ในการค้นหาข้อมูลการต่อรถเมล์ รวมถึงการค้นหาอื่น ๆ ว่ามีความเหมาะสมเพียงใดในการดำเนินงาน รวมถึงข้อดี ข้อด้อยที่เกิดขึ้น จากการศึกษาโดยใช้อัลกอริทึมแบบ อิเทอเรทีฟ ดิฟเพนนิง เซิรสช์

1.2.3 เพื่อเป็นการนำเสนอแนวคิดเปรียบเทียบว่า การค้นหาแบบใดระหว่าง เบรดท์เฟิรสท์ เซิรสช์ และ อิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ มีความเหมาะสมต่อเงื่อนไข และปัญหาที่เกิดขึ้น มากกว่ากันและจุดเด่น จุดด้อยของแต่ละวิธีทั้งคือ อัลกอริทึมแบบ เบรดท์เฟิรสท์เซิรสช์ และ อัลกอริทึมแบบ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์

1.2.4 เพื่อพัฒนาโปรแกรมระบบค้นหาเส้นทางรถเมล์ให้สมารถใช้งานได้จริง ในขอบเขตของการค้นหาที่เกิดขึ้นทั้งนี้จะเป็นการพัฒนาโดยอาศัยอัลกอริทึมที่กำหนดไว้ตามที่ได้กล่าวไว้ คือ อัลกอริทึมแบบเบรดท์เฟิรสท์เซิรสช์ และ อัลกอริทึมแบบ อิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ เพื่อให้ได้ข้อมูลตามที่ผู้ใช้ต้องการจะทำการค้นหา

1.2.5 พัฒนาส่วนการติดต่อกับผู้ใช้ คือส่วนของการรับอินพุตข้อมูล และแสดงผล ให้มีความสะดวกในการทำงานเพิ่มขึ้น โดยอยู่ในรูปของการแสดงผลบนแผนที่ เพื่อให้ผู้ใช้มีความสะดวกในการอ่านข้อมูลที่ได้จากการค้นหา และสามารถนำเอาข้อมูลที่อ่านได้ไปดำเนินการใช้ได้โดยง่าย ซึ่งต้องอาศัยอัลกอริทึมที่แสดงผลให้เหมาะสม

1.2.6 สามารถนำโปรแกรมดังกล่าวไปประยุกต์ใช้ได้กับแผนที่อื่น หรือสามารถขยายขอบเขตในการค้นหาได้โดยง่าย ซึ่งจะต้องอาศัย หลักการในอัลกอริทึม แสดงผลที่ดี มีความเป็นมาตรฐานในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผล เพื่อจะสามารถดำเนินการนำโปรแกรมประยุกต์ใช้กับแผนที่อื่น หรือขยายขอบเขตที่สนใจได้โดยสะดวก เพียงแค่เปลี่ยนข้อมูลที่รับเท่านั้น

**1.3 ขอบเขตของงานวิจัย**

จากที่กล่าวข้างต้น เราสามารถแบ่งส่วนขอบเขตของการทำงานวิจัยแบบกว้าง ๆ ในการทำงานออกเป็น 2 ส่วนคือ ส่วนของการค้นหา โดยใช้อัลกอริทึมแบบ เบรดท์เฟิรสท์เซิรสช์ และ อัลกอริทึมแบบ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ ตามที่ได้กล่าวมา และในส่วนของการติดต่อกับผู้ใช้ ซึ่งจะมีส่วนการรับอินพุตข้อมูลเพื่อนำข้อมูลส่งไปให้ส่วนการค้นหา และส่วนการรับข้อมูลจากส่วนการค้นหามาทำเป็นส่วนการแสดงผล เพื่อส่งผลลัพธ์ให้กับผู้ใช้ ซึ่งจะมีรายละเอียดต่าง ๆ ดังนี้

1.ส่วนการค้นหา โดยจะรับข้อมูลอินพุตมาจากส่วนการติดต่อกับผู้ใช้ และนำข้อมูลนั้นมาทำการหาเส้นทางที่ต่อรถเมล์น้อยที่สุด โดยใช้ทั้ง อัลกอริทึมแบบ เบรดท์เฟิรสท์เซิรสช์ และ อัลกอริทึมแบบ อิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ เมื่อหาพบก็จะส่งข้อมูลดังกล่าวคือผลลัพธ์และเส้นทางการต่อสายรถเมล์สายต่าง ๆ รวมทั้งเส้นทางรถไปให้ส่วนติดต่อกับผู้ใช้ ซึ่งการทำงานดังกล่าวนี้จะใช้อัลกอริทึมการค้นหาเป็นหลักในการทำงาน จึงจะเน้นด้านปัญญาประดิษฐ์เป็นหลัก ส่วนในการพัฒนาส่วนนี้นั้นจะใช้ จาวา แอพพลิเคชัน เป็นหลักในการพัฒนาโปรแกรมในส่วนนี้

2.ส่วนการติดต่อกับผู้ใช้ โดยจะแบ่งออกเป็น 2 ส่วนย่อย ๆ คือส่วนของการรับอินพุตข้อมูลและส่วนการแสดงผลข้อมูล ในทั้ง 2 ส่วนนี้จะมีลักษณะที่เหมือนกัน คือจะเป็นส่วนที่ใช้ภาษาจาวาแอพเพล็ตในการเขียนโปรแกรมร่วมกับภาษา เอชทีเอ็มแอล โดยในส่วนแรก คือส่วนของการรับอินพุตข้อมูล โดยจะทำการนำข้อมูลจากผู้ใช้คือจุดที่จะขึ้นและจุดหมายปลายทางส่งไปให้ส่วนแรกคือส่วนการค้นหาทำการค้นหา จากนั้นข้อมูลจะผ่านเข้าไปสู่ส่วนค้นหาค้นหาข้อมูลและส่งกลับมายังส่วนการติดต่อกับผู้ใช้อีกครั้ง หรืออาจกล่าวได้ว่า เมื่อส่วนแรกประมวลผลข้อมูลเสร็จแล้วส่วนนี้ก็จะรับข้อมูลจากส่วนแรกมาแสดงผลให้ผู้ใช้ได้เห็นในรูปของแผนที่และภาพ

นอกจากนี้ ในส่วนของการติดต่อกับผู้ใช้นี้ยังมีขอบเขตของแผนที่ที่เราสนใจเนื่องจากเราจะลดขอบเขตของแผนที่ที่เราสนใจ โดยเราจะไม่สนใจแผนที่ทั้งกรุงเทพฯ เนื่องจากขอบเขตทั้งกรุงเทพฯ จะมีขนาดที่กว้างมาก ทำให้เป็นการไม่สะดวกในการทำงาน เนื่องจากโหนดที่ใช้จะมีจำนวนมาก ยากในการทำงานของผู้ใช้ จึงมีการลดขอบเขตของแผนที่ลงมาให้เหลือเพียงแค่ 9 ส่วนย่อย ซึ่งจะกินอาณาเขต ทางฝั่งตะวันออกของกรุงเทพฯ โดยถึงแม้ว่าขอบเขตที่เราสนใจจะเป็นแค่เพียงส่วนหนึ่งของกรุงเทพฯ แต่เราก็สามารถที่จะทำการเพิ่มเติมขยายขอบเขตเข้าไปเพิ่มเติมได้มากขึ้น โดยจะยังคงใช้อัลกอริทึมแสดงผลแบบเดิมได้

อีกทั้งในส่วนของรถเมล์เราได้ทำการจำลองนำมาเพียงบางสาย เนื่องจากขอบเขตที่เรานำมาศึกษาได้ถูกตีให้แคบลง จึงจะมีรถเมล์เพียงบางส่วนจากทั้งหมดที่อยู่ในขอบเขตที่เราศึกษา และอีกทั้งในจำนวนรถเมล์บางส่วนที่เราสนใจนั้น ยังมีรถเมล์ที่วิ่งซ้ำกันอีกด้วย ในระยะทางจากจุดเริ่ม ถึงจุดสิ้นสุดในแผนที่ เช่น รถสาย 206 กับ ปอ.206 จะวิ่งบนเส้นทางเดียวกันตลอด เราจึงคัดมาเพียงแค่สายเดียวเท่านั้น เพื่อไม่ให้เป็นการซ้ำซ้อนกันของการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีขอบเขตในการแสดงผลอื่น ๆ อีก แต่จะได้กล่าวถึงต่อไป เมื่อเข้าสู่เนื้อหาการแสดงผลในบทหลัง ๆ

**1.4 วิธีการดำเนินงาน**

งานวิจัยในโครงงานนี้จะเริ่มด้วยการศึกษาทฤษฎีพื้นฐานต่าง ๆ ที่เกี่ยวข้องกับงานวิจัยซึ่งก็มีเรื่องหลัก ๆ อยู่ 2 เรื่องด้วยกัน คือ ภาษาจาวา และ โครงสร้างข้อมูล ที่ใช้ คือในส่วนของบทที่ 2 และบทที่3 จากนั้นก็จะเริ่มเข้าสู่ขั้นตอนของการออกแบบโปรแกรม โดยจะเริ่มในบทที่ 4 ที่เกี่ยวข้องกับการออกแบบโดยใช้ภาษา ยูเอ็มแอล จากนั้นจึงจะเริ่มเข้าสู่ขั้นตอนของการพัฒนาโปรแกรมอย่างแท้จริง โดยบทที่ 5 จะกล่าวถึงอัลกอริทึมที่ใช้ในการค้นหา ทั้งเบรดท์เฟิรสท์เซิรสช์ และ อิเทอเรทีฟ ดีพเพ็นนิ่งเซิรสช์ รวมทั้งองค์ประกอบโดยรวมของอัลกอริทึมที่พัฒนาขึ้นมาทั้งหมด และบทที่ 6 จะอธิบายไปถึงรูปแบบการติดต่อกับผู้ใช้ ทั้งหมดทั้งในส่วนการอินพุตข้อมูล ส่วนการแสดงผล และการประมวลผลเบื้องต้นก่อนที่จะส่งไปให้กับส่วนประกอบต่าง ๆ จากนั้นจะอธิบายแต่ละส่วน โดยในส่วนบทที่ 7 จะกล่าวถึงหลักการในการแสดงผลข้อมูล ว่ามีหลักการทำอย่างไร

โดยมีรายละเอียดในบทที่ 8 ก็จะเป็นตัวอย่างการทำงานของโปรแกรมและผลของการทดสอบอัลกอริทึมทั้งสองวิธี และบทที่ 9 ซึ่งเป็นบทสุดท้ายก็จะเป็นการสรุปการทำงาน และผลที่ได้รับจากงานวิจัยชิ้นนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ภาษา จาวา

เนื่องจากในโครงงานชิ้นนี้ จะมีการใช้ภาษาจาวาในการเขียนโปรแกรมในแต่ละส่วนจึงขอพูดถึงภาษาจาวาอย่างคร่าว ๆ ดังนี้

**ภาษาจาวา**

บริษัท ซัน ไมโครซิสเต็ม(Sun Microsystems) ทราบว่าตลาดเครื่องใช้ไฟฟ้าที่เป็นสิ่งอำนวยความสะดวกทั้งที่ใช้ส่วนตัว ภายในบ้านและในสำนักงาน เป็นตลาดที่ใหญ่กว่าตลาดคอมพิวเตอร์มากมายหลายเท่า อีกทั้งคอมพิวเตอร์และระบบสื่อสารก็ถูกย่อส่วนลงไปอยู่ในเครื่องใช้ไฟฟ้าขนาดเล็กมากขึ้นเรื่อย ๆ พวกเขาจึงคิดว่าควรต้องรีบพัฒนาเทคโนโลยี เพื่อจะทำให้บริษัทเข้าสู่ตลาดสินค้าประเภทนี้อย่างเร็วที่สุด ดังนั้นในราวปี 1991 จึงจัดทีมกรีนกรุ๊ป (Green Group) โดยมี เจมส์ โกสลิ่ง เป็นหัวหน้าเพื่อพัฒนาระบบซอฟต์แวร์สำหรับควบคุมเครื่องใช้ไฟฟ้าขนาดเล็ก อย่างเช่น โทรศัพท์มือถือ และเครื่องใช้ภายในบ้านที่มี ไมโครโปรเซสเซอร์(microprocessor) ควบคุม ในตอนแรกพวกเขาเลือกใช้ภาษา ซีพลัสพลัส (C++) ในการเขียนโปรแกรม แต่ต่อมาพบว่าภาษา ซีพลัสพลัส ไม่เหมาะสมกับการใช้งานประเภทนี้ ด้วยข้อจำกัดหลายอย่าง คือ เครื่องใช้ไฟฟ้าขนาดเล็กมักจะมีหน่วยความจำน้อยไม่สามารถทำงานโปรแกรมภาษาซีพลัสพลัส ที่มีขนาดใหญ่เพราะต้องคอมไพล์และ ลิงค์ทุก ๆ ไลบรารี ที่เกี่ยวข้องเข้าไปในไฟล์เดียว อีกทั้งในเครื่องเหล่านี้ไม่มีระบบปฏิบัติการคอยรองรับ โปรแกรมจึงไม่สามารถเรียกใช้บริการของระบบปฏิบัติการได้เหมือนบนคอมพิวเตอร์เครื่องใหญ่ ทำให้ความสามารถของภาษาจำกัดไปอย่างมาก นอกจากนี้ ซีพลัสพลัส ยังเป็นภาษาที่ไม่ปลอดภัยสำหรับเขียนโปรแกรมควบคุม เนื่องจากยอมให้มีการใช้พอยท์เตอร์ (pointer) อย่างไม่มีข้อจำกัดและละเลยการตรวจสอบชนิดของข้อมูล อย่างเคร่งครัด ปัญหาที่สำคัญกว่า คือ หน่วยประมวลผลที่ใช้ในงานควบคุมนั้นมีมากมายหลายเบอร์หลายยี่ห้อและมีชุดคำสั่งไม่เหมือนกัน โปรแกรมที่เคยทำงานบนหน่วยประมวลผลรุ่นหนึ่งจะต้องถูกคอมไพล์ใหม่จึงจะนำไปใช้งานบนหน่วยประมวลผลอีกรุ่นหนึ่งได้

ทีมงานนี้พัฒนาภาษาขึ้นมาใหม่ และตั้งชื่อว่าภาษา โอ๊ค (Oak) โดยนำเทคนิคการคอมไพล์โปรแกรมเป็นคำสั่งของหน่วยประมวลผลสมมติตัวหนึ่ง แล้วสร้างอินเตอร์พรีเตอร์ (interpreter) ของหน่วยประมวลผลสมมติตัวนั้น ให้แก่หน่วยประมวลผลตัวที่จะทำงานโปรแกรมนั้น ด้วยวิธีนี้ โปรแกรมที่สร้างขึ้นจึงสามารถนำไปทำงานบนเครื่องที่มีหน่วยประมวลผลต่างรุ่นได้ ทำให้โปรแกรมไม่ผูกติดอยู่กับหน่วยประมวลผลรุ่นใดรุ่นหนึ่ง เราเรียกคุณสมบัติอย่างนี้ว่า แพล็ตฟอร์ม อินดีเพ็นเดนท์ (platform independent)

ในปี 1992 บริษัท ซัน ไมโครซิสเต็มส์ ก่อตั้งบริษัทลูกชื่อว่า เฟิรสท์เพอซัน (FirstPerson Inc.) เพื่อร่วมมือกับบริษัท ทาม วอร์เนอร์ (Time Warner) พัฒนาระบบ วีดีโอ ออน ดีมานด์ (video on demand) ซึ่งต้องการอุปกรณ์ควบคุมเครื่องรับโทรทัศน์และติดต่อสื่อสารกับผู้ใช้ในลักษณะของ อินเตอร์แอกทีฟทีวี (interactive TV) ภาษา โอ๊ค ถูกใช้เขียนโปรแกรมของ เซ็ท-ท็อป-บ็อกซ์ (set – top – box) ซึ่งเป็นกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ต่อพ่วงกับเครื่องทีวี เพื่อควบคุมและติดต่อกับผู้ใช้ แม้ระบบนี้ถูกพัฒนาจนสามารถใช้งานได้ แต่บริษัท ทาม วอร์เนอร์ หยุดโครงการณ์นี้ไปด้วยเหตุผลที่ว่าผู้ชมและระบบเครือข่ายยังไม่พร้อม คนของ ซัน ไมโครซิสเต็มส์ จึงพยายามหาลูกค้ารายใหม่ให้แก่เทคโนโลยีนี้ โดยนำไปใช้ในงานอย่างเช่น ระบบควบคุมในบัตรเครดิต ระบบควบคุมเครื่องจักรในโรงงานอุตสาหกรรม หรือ แม้แต่ระบบควบคุมในเครื่องซีดี-รอม แต่หาลูกค้าไม่ได้สักราย เป็นที่น่าแปลกใจว่า พวกเขาเป็นแนวหน้าอยู่ในตลาดระบบเครือข่าย แต่กลับไม่ได้นึกถึงอินเตอร์เนต อยู่เป็นเวลานานนับปี

ในปี 1993 เมื่อ เอชทีเอ็มแอล และ บราวเซอร์ เปลี่ยนแปลง อินเตอร์เนต ให้ใช้ได้ง่ายและมีผู้ใช้มากขึ้นอย่างรวดเร็ว บริษัท ซัน ไมโครซิสเต็มส์ มองเห็นความจำเป็นที่จะต้องมีภาษาที่สามารถสร้างโปรแกรมบนเครื่องคอมพิวเตอร์ระบบหนึ่งแล้วส่งไปทำงานบนเครื่องต่างระบบได้ และเพิ่งนึกถึงคุณสมบัติ แพล็ตฟอร์ม อินดีเพ็นเดนท์ ของภาษา โอ๊ค ได้ จึงนำมาปัดฝุ่นใหม่และพัฒนาให้เป็นภาษาเชิงวัตถุ จนกระทั่งต้นปี 1995 ก็ตั้งชื่อใหม่ว่า จาวา (Java)

ภาษาจาวาถูกออกแบบขึ้นโดยมีจุดมุ่งหมายดังต่อไปนี้

1. เป็นภาษาที่ง่าย (simple) หมายถึง มีกลไกของภาษาจำนวนไม่มาก และไม่ซับซ้อน เพื่อให้ผู้ใช้สามารถเรียนรู้ได้อย่างรวดเร็ว ภาษาจาวานำไวยากรณ์ภาษาส่วนใหญ่มากจากภาษา ซีพลัสพลัส เพื่อให้ผู้ที่คุ้นเคยกับภาษา ซี หรือ ซีพลัสพลัส อยู่แล้วไม่ต้องเสียเวลาศึกษาไวยากรณ์ส่วนใหญ่ของภาษาจาวา จากความคิดที่ว่าภาษาที่ง่ายควรจะมีขนาดเล็ก ดังนั้นแม้ว่าภาษาจาวาจะนำไวยากรณ์มาจากภาษา ซีพลัสพลัส แต่ก็ตัดกลไกของภาษา ซีพลัสพลัส ออกไปหลายอย่าง โดยไปเพิ่มความสามารถให้คอมไพเลอร์จัดการให้แทน เช่น ภาษาจาวาไม่มี คำสั่งพรีโพรเซสเซอร์ (preprocessor commands), ฟังก์ชันโพรโตไทป์ (function prototype) และ ไฟล์ส่วนหัว (header files) รวมทั้งกลไกที่จะทำให้ภาษายุ่งยากเกินไปอย่างเช่น ดีฟอลท์ อากิวเมนต์ (default arguments) และ โอเปอเรเตอร์ โอเวอร์โหลดดิ่ง (operator overloading) ก็ถูกตัดทิ้งไปด้วย และในเมื่อจาวาเป็นภาษาเชิงวัตถุแล้ว โครงสร้างอย่างเช่น โครงสร้าง (structures), ยูเนี่ยน (unions), บิต ฟิลด์ (bit fields) และ อีนัมเมอเรต ไทป์ (enumerated types) รวมทั้งทำการทำ ไทป์เดฟ (typedef) ก็ไม่มีความจำเป็น จึงถูกตัดออกไป ภาษาจาวาถูกออกแบบให้เป็นภาษาเชิงวัตถุอย่างรอบคอบกว่าภาษา ซีพลัสพลัส ดังจะเห็นได้ว่ากลไกที่อาจทำให้เกิดความกำกวมอย่างเช่น มัลติเพิล อินเฮริแทนซ์ (multiple inheritance) และกลไกที่อาจจะทำลายแบบแผนการเขียนโปรแกรมเชิงวัตถุที่ดี อย่างเช่น เฟรินด์ เมธอด (friend methods) และ โกทู สเตทเมนต์ (goto statement) ก็ถูกตัดออกไปด้วย
2. โปรแกรมที่ถูกสร้างขึ้นด้วยภาษาจาวาจะต้องมีความผิดพลาดที่ไม่คาดคิดมาก่อนน้อยที่สุดและโปรแกรมจะต้องไม่ล้มเหลวลงง่าย ๆ ด้วยความผิดปกติเพียงเล็กน้อย คุณสมบัตินี้ เรียกว่า ความคงทน (robust) ภาษาจาวาเป็นภาษาที่เน้นการใช้กลไก เอกเซปชัน แฮนดลิง (exception handling) ซึ่งจะช่วยให้โปรแกรมสามารถจัดการกับความผิดปกติบางอย่างที่อาจเกิดขึ้นในขณะโปรแกรมทำงานโดยโปรแกรมไม่ต้องล้มเหลวและหยุดลง อย่างไรก็ตามภาษาจาวาไม่สามารถเป็นภาษาที่คงทนได้ด้วยการมี เอกเซปชัน แฮนดลิง เพียงอย่างเดียว ดังนั้นในการออกแบบภาษาจึงตัดกลไกของภาษาบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างที่อาจจะทำให้เกิดความผิดพลาด (หากใช้อย่างไม่ระวัง) ออกไปและเพิ่มการตรวจสอบว่าการทำงานบางอย่างจะก่อให้เกิดความผิดพลาดขึ้นได้หรือไม่

ภาษาจาวาตัดการอ้างถึงแอดเดรสของตัวแปร และการใช้พอยท์เตอร์ ในการอ่านหรือเขียนค่าหน่วยความจำโดยตรงออกไปจากภาษา เพื่อให้โปรแกรมไม่มีโอกาสใช้พอยท์เตอร์อ้างถึงหน่วยความจำที่โปรแกรมไม่มีสิทธิ์อ้างถึง การสร้างโปรแกรมในลักษณะนี้จึงเป็นไปไม่ได้นอกจากนี้ภาษาจาวายังกำหนดให้ ผู้โปรแกรมไม่สามารถควบคุมการคืนหน่วยความจำ (memory deallocation) ในขณะโปรแกรมทำงาน แต่อาศัย ออโตเมติก กาเบกจ์ คอลเล็กเตอร์ (automatic garbage collector) ทำหน้าเก็บหน่วยความจำที่ไม่อาจอ้างถึงได้แล้ว และนำกลับมาใช้งานอีกโดยอัตโนมัติโปรแกรมจึงไม่มีโอกาสอ้างถึงค่าในหน่วยความจำที่ถูกทำลายไปแล้ว

ภาษาจาวาถูกจัดว่าเป็น ภาษาที่มีการจัดการชนิดของข้อมูลอย่างรัดกุม (strongly typed language) ซึ่งหมายถึง ภาษาที่เน้นความถูกต้องของชนิดข้อมูล (type) ที่ใช้ในโปรแกรม คอมไพเลอร์ของภาษาประเภทนี้จะทำการตรวจสอบการจัดการกับชนิดข้อมูลของตัวแปรว่าถูกต้องหรือไม่ ซึ่งเรียกว่า การตรวจสอบชนิดของข้อมูล ความผิดพลาดเกี่ยวกับชนิดข้อมูลจะถูกปฏิเสธตั้งแต่ตอนคอมไพล์โปรแกรม ทำให้ไม่มีความผิดพลาดเกี่ยวกับชนิดของข้อมูลเกิดขึ้นในระหว่างที่โปรแกรมทำงาน ถือเป็นภาษาที่ปลอดภัยในระดับหนึ่ง แต่ก็เป็นภาษาที่ไม่มีความยืดหยุ่นเกี่ยวกับชนิดข้อมูล ภาษาจาวามีคอมไพเลอร์มีการตรวจสอบชนิดของข้อมูล อย่างเคร่งครัด และยังมีการตรวจสอบอีกหลายอย่างในระหว่างที่โปรแกรมทำงานอีกด้วย เช่นตรวจสอบว่าการเปลี่ยนชนิดข้อมูล (casting) ว่าถูกต้องหรือไม่ และการอ้างถึงสมาชิกในอาร์เรย์ หรือสตริงอยู่ในขอบเขตที่ถูกต้องหรือไม่

3. โปรแกรมที่สร้างขึ้นด้วยภาษาจาวา มักจะถูกส่งผ่านระบบเครือข่ายไปทำงานในเครื่องคอมพิวเตอร์ของผู้อื่น ดังนั้นภาษาจาวา ต้องมีหลักประกันแก่ผู้รับโปรแกรมไปทำงานว่า มันจะไม่ก่อให้เกิดอันตรายต่อเครื่องหรือระบบข้อมูลของผู้ใช้ ในการออกแบบภาษาจาวาจึงมีข้อกำหนดหลายอย่างที่จะทำให้ไม่สามารถสร้างโปรแกรมเพื่อส่งไปทำอันตราย (หรือสิ่งที่ไม่สมควรทำ) ต่อเครื่องที่รับโปรแกรมนั้นไปทำงาน คุณสมบัติอย่างนี้เรียกว่า ความปลอดภัย (secure) อย่างไรก็ตามไม่มีภาษาใดที่ปลอดภัยร้อยเปอร์เซ็นต์ ภาษาจาวามีความปลอดภัยในระดับสูงเพราะถูกออกแบบมาเพื่อความปลอดภัยมากกว่าภาษาอื่น โดยแบ่งการป้องกันออกเป็นหลายระดับดังนี้

- ที่ระดับล่าง ภาษาจาวาไม่อนุญาตให้อ้างถึงค่าในหน่วยความจำโดยตรงผ่านทางพอยท์เตอร์ และทำการตรวจสอบว่าการอ้างถึงสมาชิกในอาร์เรย์อยู่ในขอบเขตของอาร์เรย์นั้นหรือไม่ ทำให้ไม่สามารถเขียนหรืออ่านค่าในหน่วยความจำที่โปรแกรมนั้นไม่มีสิทธิ์ ดังนั้นการพยายามเปลี่ยนแปลงโปรแกรมหรือค่าในหน่วยความจำเพื่อสร้างโปรแกรมที่จะเป็นอันตรายต่อคนอื่นจึงเป็นไปไม่ได้
- ที่ระดับถัดขึ้นมา ในที่โปรแกรมทำงานจำมีโปรแกรมที่เรียกว่า ไบต์-โค้ด เวอริไฟเออร์ (byte-code verifier) ทำหน้าที่ตรวจสอบโปรแกรมที่จะถูกทำงานว่ามีรหัสคำสั่งที่ผิดปกติหรือมีการกระทำใดที่ไม่สมควรหรือไม่ หากพบการกระทำใดที่ก่อให้เกิดอันตรายต่อระบบก็จะปฏิเสธที่จะทำงานโปรแกรมนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่ระดับบน มีระบบรักษาความปลอดภัยที่เรียกว่า แซนด์บ็อกซ์ โมเดล (sandbox model) นั่นคือ โปรแกรมที่ถูกดาวน์โหลดมากจากเครื่องอื่นจะถูกจัดว่าเป็นโปรแกรมที่ไม่น่าไว้ใจ (unstrusted codes) และถูกเก็บอยู่ใน แซนด์บ็อกซ์ โปรแกรมที่อยู่ใน แซนด์บ็อกซ์ นี้จะมีข้อจำกัดในการทำงานหลายอย่างซึ่งถูกควบคุมโดยตัวจัดการเกี่ยวกับความปลอดภัย (Security Manager) เช่น ไม่สามารถอ่านหรือเขียนไฟล์ เป็นต้น

อีกจุดมุ่งหมายหนึ่งของการออกแบบภาษาจาวา คือการทำให้โปรแกรมที่สร้างขึ้นถูกส่งไปทำงานบนเครื่องต่างชนิดได้ ซึ่งถือเป็นเรื่องที่สำคัญที่สุด

ภาษาระดับสูงจะต้องใช้ตัวแปลภาษา (language translator) เพื่อเปลี่ยนโปแกรมต้นฉบับ (sc irce code)ให้เป็นโปรแกรมของคำสั่งที่หน่วยประมวลผลทำงานได้ (machine code) โดยทั่วไปวิธีแปลภาษาจะถูกแบ่งออกเป็นสองแบบคือ

1. การคอมไพล์ (Compilation) ตัวแปลภาษาในกรณีนี้เรียกว่า คอมไพเลอร์ มีหน้าที่วิเคราะห์โปรแกรม ซอร์ซ โค้ด (source code) แล้วสร้างโปรแกรม แมชีน โค้ด (machine code) ออกมาเป็นผลลัพธ์ การกระทำดังกล่าวเรียกว่าการคอมไพล์โปรแกรม ข้อดีของวิธีนี้คือ โปรแกรมแมชีนโค้ด ทำงานเร็วมาก เนื่องจากขั้นตอนในการแปลภาษาถูกแยกออกไปทำก่อนโปรแกรมการทำงาน และคอมไพเลอร์ก็สามารถวิเคราะห์โปรแกรมได้ทั้งโปรแกรมเสียก่อน จึงสร้างแมชีนโค้ด ที่มีประสิทธิภาพในการทำงานและโปรแกรมมีขนาดเล็ก ตัวอย่างของภาษาที่ใช้การแปลภาษาแบบคอมไพเลชัน (compilation) คือ ฟอร์แทรน (Fortran), อัลกอล (Algol), ปาสคาล (Pascal), ซี, และ ซีพลัสพลัส
2. อินเตอร์พรีเตชัน (Interpretation) ตัวแปลภาษาในกรณีนี้เรียกว่า อินเตอร์พรีเตอร์ มีหน้าที่อ่านโปรแกรม ซอร์ซโค้ด ทีละบรรทัด และแปลโปรแกรมบรรทัดนั้นเป็น แมชีนโค้ด และทำงานทันที ต่อจากนั้นก็อ่านโปรแกรม ซอร์ซโค้ด บรรทัดต่อไปเข้ามาทำเช่นเดิมอีก ไปจนกว่าโปรแกรมจะหยุดหรือจบโปรแกรม การกระทำดังกล่าวเรียกว่า การ อินเตอร์พรีต (interpret) วิธีนี้จะมีทั้งการแปลภาษาและโปรแกรมทำงานเกิดขึ้นสลับกันไป โปรแกรมจึงทำงานช้ากว่าวิธีคอมไพล์ แต่ก็มีข้อดีคือ อินเตอร์พรีเตอร์ ถูกสร้างขึ้นได้ง่ายและมีขนาดเล็กกว่า เพราะการวิเคราะห์และแปลโปรแกรมทีละบรรทัดทำได้ง่ายกว่าแปลทั้งโปรแกรม และการแปลภาษาระหว่างโปรแกรมทำงานจะทำให้ได้ภาษาที่มีความยืดหยุ่นในการเขียนโปรแกรมมากกว่า ตัวอย่างของภาษาที่ใช้การแปลภาษาแบบ อินเตอร์พรีเตชัน คือ ลิสป์ (Lisp), เบสิก (Basic), โปรลอก (Prolog) และ สมอลทอลค์ (Smalltalk)

วิธีแปลภาษาทั้งสองแบบมีทั้งข้อดี และข้อเสียที่ทำให้โดยปกติภาษาหนึ่งจะเลือกใช้การแปลภาษาเพียงแบบใดแบบหนึ่งเท่านั้น แต่ก็มีบางภาษาที่นำเอาการแปลภาษาทั้งสองแบบมาใช้เพื่อประโยชน์ ดังตัวอย่างต่อไปนี้

ในต้นยุค 1970s นักวิจัยที่ ยูซี ซาน ดิเอโก (UC San Diego) สร้างคอมไพเลอร์ที่แปลโปรแกรมภาษาปาสคาล ไปเป็นโปแกรมของภาษาสมมติที่เรียกว่า พี-โค้ด (p-code) แล้วโปรแกรมพี-โค้ด จะถูกทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย อินเตอร์พรีเตอร์ อีกต่อหนึ่ง เหตุที่ต้องทำเช่นนี้ก็เพราะ ในสมัยนั้นยังไม่มีคอมไพเลอร์ที่แปลจากภาษาปาสคาล ไปเป็นภาษาเครื่องทันที ถือว่าเป็นเรื่องที่ยากมาก อีกทั้งในเวลานั้นยังไม่มีบริษัทคอมพิวเตอร์ใดครองตลาดได้อย่างแท้จริง มีหน่วยประมวลผลหลากหลายยี่ห้อและแตกต่างกันในเครื่องต่าง ๆ หากจะสร้างคอมไพเลอร์ภาษาปาสคาล ให้แก่หน่วยประมวลผลรุ่นหนึ่ง ก็อาจไม่คุ้มค่า เพราะใช้ได้ไม่แพร่หลาย ดังนั้นเขาจึงใช้วิธีคอมไพล์จากภาษาปาสคาล ไปเป็นภาษาสมมติ พี-โค้ด ซึ่งเป็นภาษาระดับสูงกว่าเครื่อง จึงทำได้ง่ายกว่าการคอมไพล์ไปเป็นภาษาเครื่อง และสร้างอินเตอร์พรีเตอร์ของภาษาพี-โค้ด สำหรับหน่วยประมวลผลแต่ละเบอร์ ซึ่งก็ไม่ใช่งานที่ยากนัก เพราะภาษาพี-โค้ด นั้นเกือบเหมือนกับภาษาเครื่องแล้ว วิธีนี้ไม่เป็นที่แพร่หลายเพราะโปรแกรมทำงานช้า และเมื่อเข้าสู่ยุคที่หน่วยประมวลผลของบริษัท อินเทล (Intel) และ โมโตโรลา (Motorola) ได้รับความนิยมอย่างมาก ความหลากหลายของหน่วยประมวลผลจึงลดลง และการสร้างคอมไพเลอร์ไปสู่ภาษาเครื่องของหน่วยประมวลผลที่มีคนนิยมใช้ จึงประสบความสำเร็จมากกว่า

ต่อมาในปลายยุค 1970s ภาษาสมอลทอล์ค พัฒนาความคิดนี้ไปอีกแนวหนึ่งคือ แทนจะกำหนดเพียงภาษาสมมติเท่านั้นแต่กำหนดไปถึงโครงสร้างและพฤติกรรมของเครื่องจักรสมมติ (virtual machine) สำหรับทำงานโปรแกรมภาษาสมมตินั้นด้วย ภาษาสมอลทอล์ค ใช้การแปลภาษาแบบอินเตอร์พรีเตชัน โดยอินเตอร์พรีเตอร์ทำหน้าที่แปลโปรแกรมทีละบรรทัดไปเป็นโปรแกรมภาษาสมมติแล้วจะถูกทำงานโดยเครื่องจักรสมมติที่จำลองขึ้น ซึ่งถูกกำหนดเป็นมาตรฐานในทุก อินเตอร์พรีเตอร์ ของเครื่องทุกเครื่องคอมพิวเตอร์ไม่ว่าจะใช้หน่วยประมวลผลใด วิธีการนี้ทำให้โปรแกรมภาษา สมอลทอล์ค ทำงานได้บนเครื่องคอมพิวเตอร์ใด ๆ ที่มี สมอลทอล์ค อินเตอร์พรีเตอร์ โดยมีผลลัพธ์เหมือนกันหมด และไม่ต้องเปลี่ยนแปลงโปรแกรมส่วนใดเลยเพื่อจะไปทำงานบนอีกเครื่องหนึ่ง แม้ภาษาสมอลทอล์คจะประสบความสำเร็จในเรื่องนี้ แต่ก็ยังติดปัญหาที่โปรแกรมทำงานช้ามาก

**2.2 จาวาเวอร์ชวลแมชีน (Java's Virtual Machine)**

ภาษาจาวานำความคิดในการสร้างเครื่องจักรสมมติมาใช้ เพื่อให้ภาษามีคุณสมบัติ แพล็ตฟอร์มอินดีเพ็นเดนท์ (platform independent) แต่แยกการแปลภาษาออกไปโดยใช้คอมไพเลอร์ แล้วนำโปรแกรมของเครื่องจักรสมมติที่ได้มาทำงานโดย อินเตอร์พรีเตอร์ วิธีการนี้จะทำให้โปรแกรมทำงานได้เร็วขึ้นกว่าการใช้ อินเตอร์พรีเตอร์ เพียงอย่างเดียว

ไฟล์ของโปรแกรม ซอร์ซโค้ด ภาษาจาวาจะต้องมีส่วนขยาย (extension) เป็น .java คอมไพเลอร์ของ ภาษาจาวาจะทำหน้าที่คอมไพล์โปรแกรมภาษาจาวาไปเป็นโปรแกรมที่ประกอบด้วยคำสั่งของจาวาเวอร์ชวลแมชีน (Java Virtual Machine (JVM)) และจะถูกเก็บในไฟล์ที่มีส่วนขยาย เป็น .class เรียกว่าไฟล์จาวาคลาส (Java class) ด้วยเหตุที่โปรแกรมจาวาคลาส ต้องถูกทำงานด้วยเจวีเอ็ม (JVM) ดังนั้นบางทีเราจึงเรียกเจวีเอ็ม ว่า จาวารันทามซิสเต็ม (Java runtime system)

ปัจจุบันบริษัท จาวาซอฟต์ (JavaSoft) ซึ่งเป็นส่วนหนึ่งของบริษัท ซัน ไมโครซิสเต็มส์ ได้กำหนดมาตรฐานของภาษาจาวา และเป็นผู้กำหนดชุดคำสั่งของเจวีเอ็ม รวมทั้งความหมายของแต่ละคำสั่ง ข้อกำหนดเหล่านี้เป็นมาตรฐานของภาษาจาวา ซึ่งเผยแพร่ให้แก่บุคคลทั่วไป ซึ่งใครก็อาจจะสร้างเจวีเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของตนเองขึ้นได้ โดยอาจจะเป็นฮาร์ดแวร์ หรือ ซอฟต์แวร์ก็ตาม แต่ต้องสามารถทำงานโปรแกรมจาวาคลาส ให้ได้ผลลัพธ์ตามที่มาตรฐานกำหนดไว้

ภายในเจวีเอ็ม มีหน่วยประมวลสมมติที่เรียกว่า เวอร์ชวลโพรเซสเซอร์ (virtual processor) ทำหน้าที่ประมวลคำสั่งของเจวีเอ็มปัจจุบัน เจวีเอ็ม ที่เป็นฮาร์ดแวร์ยังอยู่ในระหว่างการพัฒนา ดังนั้น เจวีเอ็ม เกือบทั้งหมดที่ใช้กันอยู่ในตอนนี้จึงเป็นโปรแกรมที่จำลองการทำงานของ เจวีเอ็ม บนเครื่องคอมพิวเตอร์ทั่วไป โดยทั่วไปเวอร์ชวลโพรเซสเซอร์ของเจวีเอ็มที่จำลองการทำงานของเจวีเอ็มบนเครื่องคอมพิวเตอร์เครื่องหนึ่ง จะแปลคำสั่งของเจวีเอ็มเป็นคำสั่งของหน่วยประมวลผลในคอมพิวเตอร์เครื่องนั้นเรียกว่า เนทีฟโค้ด (native code) แล้วให้หน่วยประมวลผลนั้นทำงานคำสั่งนั้น คำสั่ง (opcode) ของ เจวีเอ็ม มีขนาด 1 ไบต์ ทุกคำสั่ง (บางครั้งเราจึงเรียกโปรแกรมจาวาคลาส ว่าโปรแกรม byte code) จำนวนคำสั่งของเจวีเอ็มจึงมีได้สูงสุดเพียง 256 คำสั่ง เปรียบกับหน่วยประมวลผลทั่วไปแล้วคล้ายกับว่าคำสั่งของเจวีเอ็ม มีมากมายอย่างยิ่ง แต่จริง ๆ แล้วคำสั่งของเจวีเอ็มแบ่งออกได้เป็นไม่กี่ประเภท โดยที่แต่ละประเภทนั้นจะทำหน้าที่คล้าย ๆ กัน เพียงแต่ทำกับโอเปอแรนด์ (operands) ต่างชนิดข้อมูลกันเท่านั้น

ชุดคำสั่งของเจวีเอ็มถูกออกแบบมาเพื่อสนับสนุนการทำงานของโปรแกรมเชิงวัตถุจึงคำสั่งเกี่ยวกับการสร้างอินสแตนซ์ (instances) และการอ้างถึงสมาชิกในอินสแตนซ์ ซึ่งไม่มีในหน่วยประมวลผลทั่วไป ภาษาจาวาเป็นภาษาที่เน้นความถูกต้องเกี่ยวกับชนิดของข้อมูล (type) จึงมีคำสั่งสำหรับคำนวณชนิดข้อมูลพื้นฐานแต่ละชนิดเช่น คำสั่ง iadd สำหรับบวกเลขจำนวนเต็มอินทีเจอร์ (integer) และคำสั่ง dadd สำหรับบวกเลขทศนิยมดับเบิล (double) เป็นต้น บางคำสั่งของเจวีเอ็มจะเหมือนกับคำสั่งที่มีในหน่วยประมวลผลทั่วไป แต่ก็มีหลายคำสั่งที่ไม่มีในหน่วยประมวลผลทั่วไป จึงต้องสร้างขึ้นเป็นโปรแกรมของคำสั่งนั้น

เจวีเอ็ม ถูกออกแบบให้สามารถจำลองได้บนเครื่องทั่วไปไม่ว่าจะใช้หน่วยประมวลผลของบริษัทใด แต่หน่วยประมวลผลต่างรุ่นต่างบริษัทมีจำนวนรีจิสเตอร์ (registers) ไม่เท่ากัน บางรุ่นมีรีจิสเตอร์หน้าที่พิเศษที่รุ่นอื่นไม่มี ผู้ออกแบบจึงตัดปัญหานี้โดยให้เจวีเอ็มไม่มีรีจิสเตอร์และทำการคำนวณทั้งหมดบนสแตก (stacks) ชุดคำสั่งของเจวีเอ็มจึงเป็นสแตกโอเปอเรชัน (stacked operations) หรือกล่าวได้ว่าเจวีเอ็มเป็นสแตกแมชีน (stack machine)

เวอร์ชวลโพรเซสเซอร์ในเจวีเอ็มจะทำการแมปปิ้ง (mapping) จากคำสั่งไบต์โค้ด (byte code) ในไฟล์จาวาคลาสไปเป็นโปรแกรมของเนทีฟโค้ดที่ทำหน้าที่ของคำสั่งนั้น (โดยอาจจะใช้แฮชเทเบิล (hash table) หรือสร้างขึ้นโดยประโยคสวิตช์ (switch) ในภาษาซี) แล้วส่งเนทีฟโค้ด นั้นให้หน่วยประมวลผลทำงาน สังเกตว่าเนทีฟสแตนดาร์ดคลาส (native standard classes) ที่สร้างขึ้นสำหรับหน่วยประมวลผลนั้นทำให้เจวีเอ็มหนึ่งอาจใช้งานได้ในระบบต่าง ๆ โดยเปลี่ยนแปลงแต่สแตนดาร์ดคลาส (standard classes) เท่านั้น

แน่นอนว่าโปรแกรมที่ทำงานโดยอินเตอร์พรีเตอร์ ย่อมช้ากว่าโปรแกรมที่ทำงานโดยตรง ปัจจุบันโปรแกรมภาษาจาวาทำงานช้ากว่าโปรแกรมภาษาซี หรือ ซีพลัสพลัส โดยเฉลี่ยประมาณ 20 เท่า แต่ก็เร็วกว่าภาษาเบสิก, สมอลทอล์ค, หรือ จาวาสคริปต์ (JavaScript) มาก เพราะอินเตอร์พรีเตอร์ของภาษาจาวาไม่ต้องเสียเวลาแปลจากภาษาระดับสูงในระหว่างที่โปรแกรมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันมีการค้นคว้าหลายอย่างเพื่อเพิ่มความเร็วของภาษาจาวา เช่นการสร้างหน่วยประมวลผลที่สามารถทำงานคำสั่งของเจวีเอ็มได้โดยตรง หรือพัฒนาอินเตอร์พรีเตอร์ซึ่งไม่ทำงานคำสั่งของเจวีเอ็มทีละคำสั่ง แต่จะเปลี่ยนโปรแกรม จาวาคลาส ทั้งโปรแกรมเป็นเนทีฟโค้ดแล้วจึงทำงานเนทีฟโค้ดนั้น วิธีการนี้เรียกว่า จัสอินทาม (Just In Time (JIT)) เนื่องจากคล้ายกับทำการคอมไพล์โปรแกรมก่อนจะเริ่มทำงาน โปรแกรมจะทำงานเร็วขึ้นอย่างมาก เพราะคำสั่งที่ถูกทำซ้ำบ่อย ๆ เช่นคำสั่งที่อยู่ในประโยคทำซ้ำหรือฟังก์ชันที่ถูกเรียกใช้บ่อย ๆ จะถูกแปลเพียงครั้งเดียว ไม่ต้องถูกแปลทุกครั้งที่จะถูกทำงาน และยังทำให้นำเทคนิคการปรับปรุงโปรแกรมให้เร็วขึ้น (optimization) ที่มีใช้ในคอมไพเลอร์ทั่วไปมาใช้ได้ด้วย

**2.3 จาวาแอพเพล็ต**

แอพเพล็ต (Applet) คือโปรแกรมขนาดเล็กที่สร้างขึ้นด้วยภาษา จาวา สามารถถูกเรียกจากในเอชทีเอ็มแอล เพจ ให้ทำงานเป็นส่วนหนึ่งของ เว็บเพจ นั้น กล่าวอีกอย่างได้ว่าเป็นโปรแกรมที่ถูกส่งไปกับ เอชทีเอ็มแอลเพจ เพื่อไปทำงานภายใต้ บราวเซอร์ที่มีจาวาอินเตอร์พรีเตอร์ (Interpreter) อยู่บนเครื่องที่รับเอชทีเอ็มแอลเพจนั้นไป เป็นที่สังเกตได้ว่าโปรแกรมภาษาจาวาที่เป็น แอพพลิเคชัน จะทำงานภายใต้จาวาอินเตอร์พรีเตอร์ โดยไม่จำเป็นต้องอาศัยโปรแกรมอื่น จึงเรียกว่าเป็นสแตนด์อโลน (Stand Alone) โปรแกรม และสามารถที่จะควบคุมการดำเนินงานของตัวเอง แต่โปรแกรมที่เป็น แอพเพล็ต จะทำงานภายใต้ บราวเซอร์ โดยไม่สามารถควบคุมการดำเนินงานของตัวเองได้ทั้งหมด

คลาสแอพเพล็ต เก็บอยู่ใน แพคเกจ java.applet ที่มากับโปรแกรม เจดีเค (JDK) ใช้สำหรับสร้าง แอพเพล็ต โดยทุกโปรแกรมของแอพเพล็ตจะต้องมีประโยค “import java.applet.*;” เพื่อให้คอมไพเลอร์สามารถมองเห็นคลาสแอพเพล็ต และประโยค “import java.applet.*;” เพื่อให้คอมไพเลอร์มองเห็นแพคเกจ ที่เรียกว่า แอ็บสแทรกซ์ วินโดว์ ทูลคิท (Abstract Window Toolkit (AWT) ) ซึ่งเป็นที่รวบรวมของคลาสที่เกี่ยวกับ กราฟิกส์ และ กราฟิกส์ ยูสเซอร์ อินเตอร์เฟส (จียูไอ ) โดยทั่วไปเราจะไม่สร้างอินสแตนซ์ ของคลาส แอ็พเพล็ต ขึ้นใช้โดยตรง เพราะจะได้ แอ็พเพล็ต ที่ไม่ทำอะไรเลย แต่จะใช้งานคลาส แอพเพล็ตโดยขยายไปเป็นคลาสลูก โดยทำการโอเวอร์ไรด์ เมธอด ( Override Method) บางตัวให้แอพเพล็ต ที่ได้นั้นทำงานตามที่เราต้องการสังเกตว่าโปรแกรมของ แอพเพล็ต ไม่จำเป็นต้องมี main() เป็นจุดเริ่มต้นของการทำงาน แม้อาจจะมีแต่ก็จะไม่ถูกเรียก เพราะแอพเพล็ตมีช่วงชีวิต ที่ถูกควบคุมโดยบราวเซอร์

โปรแกรมของแอพเพล็ตเมื่อถูกคอมไพล์แล้วจะได้หนึ่งไฟล์เป็น .Class แต่จะไม่สามารถทำงานได้โดยจาวาอินเตอร์พรีเตอร์ การเรียกใช้ แอพเพล็ตทำงานนั้นต้องทำจากภายในเอชทีเอ็มแอล เพจ โดยใช้แอพเพล็ตแทก และแอพเพล็ตนั้นจะทำงานในสภาวะแวดล้อมของบราวเซอร์อย่างเช่น เน็ตสเคปนาวิเกเตอร์, ไมโครซอฟต์ อินเตอร์เนต เอกซ์พรอเลอร์, ฮอทจาวา ซึ่งทั้งหมดนี้จะเป็น บราวเซอร์ ที่มีจาวาอินเตอร์พรีเตอร์ อยู่ภายในหรือจะใช้โปรแกรมสำหรับทดสอบที่มีมาให้กับ เจดีเค คือ แอพเพล็ตวิว-เวอร์ (Appletviewer) โดย แอพเพล็ตวิวเวอร์นี้จะทำงานได้เร็วกว่า บราวเซอร์ทั่วไป และสนับสนุนกลไกของภาษา จาวาที่เพิ่มขึ้นใหม่อย่างครบถ้วน ถูกต้องตามมาตรฐานของภาษาที่กำหนดโดยบริษัทจาวาซอฟต์ อย่างไรก็ตาม แอพเพล็ตวิวเวอร์ นี้ไม่ได้ถูกสร้างมาเพื่อเป็น บราวเซอร์ มันจึงไม่สามารถจัดการกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความส่วนอื่น ๆ ใน เอชทีเอ็มแอล แพจ ได้ และจะทำงานเฉพาะ แอพเพล็ต เท่านั้น อีกทั้งยังไม่สามารถติดต่อกับระบบ เน็ตเวิร์กได้เหมือนกับ บราวเซอร์ทั่วไป ดังนั้นในบางกรณีเราจำเป็นต้องใช้ บราวเซอร์จริง ๆ ในการตรวจสอบการทำงานของ แอพเพล็ต บางตัว ที่ต้องการสภาวะแวดล้อมของ บราวเซอร์

ช่วงชีวิตของ แอพเพล็ต แบ่งออกเป็นสถานะต่าง ๆ ได้ดังนี้ สถานะเริ่มแรก (init) ,สถานะเริ่มต้น (start) ,สถานะวาด (paint) , สถานะรอ (wait) ,สถานะแขวน (suspended ) ,สถานะหยุด (stop) และ สถานะทำลาย (destroy) ทุกๆแอพเพล็ต จะดำเนินการผ่านสถานะเหล่านี้ โดยบางส่วนจะเป็นไปอย่างอัตโนมัติทันที และบางส่วนอยู่ในการควบคุมของผู้ใช้ที่เรียกดู แอพเพล็ต ,นั้น ช่วงชีวิตของ แอพเพล็ต แสดงเป็นรูปการเปลี่ยนสถานะได้ดังนี้

จากรูปเมื่อเริ่มต้นแอพเพล็ตถูกสร้างขึ้นโดยการ new() เหมือนอินสแตนซ์ทั่วไป ก็จะเริ่มทำงานทันทีคือ

1.เข้าสู่สถานะเริ่มแรก แล้วผ่านเข้าสู่สถานะวาด และเข้าสู่สถานะรอ ซึ่งจะหยุดรอเหตุการณ์ (Event) หรือสัญญาณจากผู้ใช้ (คือ อินพุตที่รับเข้าไปทาง เมาส์,คีย์บอร์ด หรือ กราฟิกส์ ยูสเซอร์ อินเตอร์) ที่จะเข้ามาทำให้เกิดการทำงานของเมธอดที่กำหนดไว้สำหรับจัดการกับเหตุการณ์นั้น เมื่อเสร็จแล้วจะกลับเข้าสู่สถานะรอ (wait) เพื่อรอเหตุการณ์อื่นต่อไป

2.ระหว่างที่ แอพเพล็ตอยู่ในสถานะรออยู่หากผู้ใช้เปลี่ยนแปลงเว็บเพจจนทำให้

2.1 พื้นที่แสดงผลของแอพเพล็ตนั้นต้องมีการถูกวาดใหม่ เช่นเมื่อแอพเพล็ตถูกย้ายที่ (move) เปลี่ยนขนาด (resize) ขยายขนาด (maximize) หรือมีวินโดว์อื่นมาทับแอพเพล็ตแล้วนำวินโดว์นั้นออกไป บราวเซอร์จะออกคำสั่งให้แอพเพล็ตทำการวาดอีกครั้งโดยเรียกใช้ repaint () ซึ่งจะส่งผลให้แอพเพล็ต เข้าสู่สถานะรอเพื่อรอเหตุการณ์อื่นต่อไป

2.2 พื้นที่แสดงผลของแอพเพล็ต นั้นไม่ปรากฏในเว็บเพจปัจจุบัน เช่น เมื่อผู้ใช้ทำการเลื่อนเพจไปดูหน้าอื่น หรือกดปุ่ม ย่อขนาด (minimize) จะทำให้บราวเซอร์ออกคำสั่งให้แอพเพล็ตผ่านเข้าสู่สถานะหยุด และไปสู่สถานะแขวน ซึ่งแอพเพล็ต จะหยุดรออยู่ในสถานะนี้โดยไม่ยอมรับอินพุตจากผู้ใช้และไม่โต้ตอบใด ๆ ไปจนกว่าพื้นที่แสดงผลของแอพเพล็ตนั้นจะถูกทำให้กลับมาปรากฏอีกครั้ง บราวเซอร์ ก็จะออกคำสั่งให้แอพเพล็ต นั้นเข้าสู่สถานะ เริ่มต้น,สถานะวาด, และ สถานะรอ เพื่อเริ่มทำงานอีก

3. ระหว่างที่ แอพเพล็ต อยู่ในสถานะรอ หากผู้ใช้กดปุ่มปิดของบราวเซอร์เพื่อหยุดการทำงานจะทำให้แอพเพล็ต เข้าสู่สถานะหยุดและไปสู่สถานะทำลาย แล้วสิ้นสุดการทำงาน

ข้อสังเกตเกี่ยวกับช่วงชีวิตของแอพเพล็ตพอสรุปได้ดังนี้

- เมื่อเริ่มต้นทำงานแอพเพล็ตจะเข้าสู่สถานะเริ่มแรก, สถานะเริ่มต้น และสถานะวาด
- ตอนถูกทำลายก็จะเข้าสู่สถานะหยุด และสถานะทำลาย
- แอพเพล็ตหนึ่งจะผ่านสถานะเริ่มแรก และสถานะทำลายได้ครั้งเดียวนั่นคือ ผ่านสถานะเริ่มแรก เมื่อตอนเริ่มต้นทำงานครั้งแรก และเมื่อผ่านสถานะทำลาย แล้วจะกลับมาทำงานอีกไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แอพเพล็ต หนึ่งจะผ่านสถานะเริ่มต้น หรือ สถานะหยุด ได้หลายครั้งขึ้นกับการควบคุมของผู้ใช้ที่ดู แอพเพล็ตนั้นอยู่ว่ามีการควบคุมอย่างไร
- แอพเพล็ตจะหยุดรอเหตุการณ์อยู่ในสถานะรอ หรือ สถานะแขวน ได้เป็นเวลานาน ส่วนสถานะอื่น ๆ นั้น แอพเพล็ตมีการเข้าไปทำงานแล้วจะผ่านออกไปทันที

ในการสร้างโปรแกรมของแอพเพล็ต เราสามารถกำหนดรูปแบบการทำงานในแต่ละสถานะต่าง ๆ ที่กล่าวมา เมื่อ แอพเพล็ต ดำเนินการผ่านคลาส แอพเพล็ต ก็จะมี เมธอด ชื่อเหมือนกับสถานะทั้ง 5 ใช้สำหรับ กำหนดการทำงานเมื่อตอน แอพเพล็ต นั้นเข้าสู่สถานะนั้น ๆ ถ้าตรวจสอบดูจะพบว่า paint () ไม่ได้ถูกกำหนดไว้ในคลาสแอพเพล็ต แต่มันจะได้รับการถ่ายทอดมาจากคลาสคอมโพเน็นต์ และสังเกตว่าจะไม่มีเมธอด สำหรับสถานะแขวน และ สถานะรอ เนื่องจากการเข้าสู่และออกจากสถานะแขวน นั้นอยู่ในการควบคุมของบราวเซอร์ ส่วนการโปรแกรมแอพเพล็ต ในสถานะรอ นั้นจะใช้ อีเวนท์ แฮนดลิง (Event Handling ) ในการจัดการ

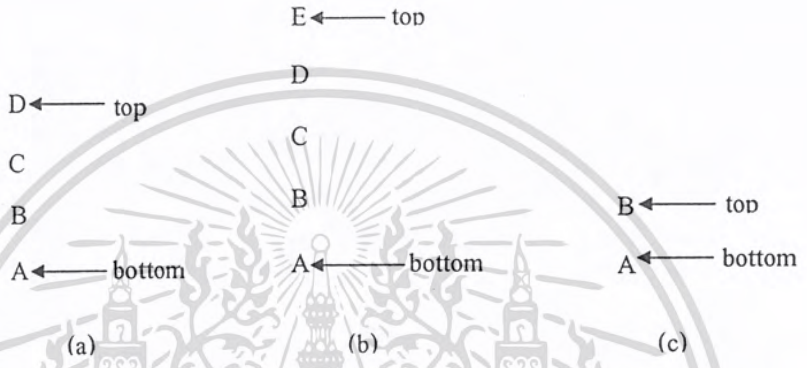
จากทฤษฎีของจาวาแอพเพล็ตข้างต้นเราสามารถนำบางส่วนมาประยุกต์ใช้เข้ากับโปรแกรมที่เขียนในส่วนการรับอินพุตข้อมูลและส่วนการแสดงผล โดยจะใช้ร่วมกันกับภาษาเอชทีเอ็มแอล ซึ่งจะเว้นไว้ไม่กล่าวถึงรายละเอียดทฤษฎีของภาษาเอชทีเอ็มแอล เนื่องจากเป็นภาษาที่ผู้ใช้ส่วนใหญ่มีความเข้าใจในหลักการเขียนภาษา เอชทีเอ็มแอลนี้พอสมควรเนื่องจากเป็นภาษาพื้นฐานของการเขียนเว็บในปัจจุบัน ไม่เหมือนกับภาษาจาวาที่เป็นภาษาที่ค่อนข้างใหม่สำหรับผู้ใช้ทั่วไป จึงต้องกล่าวมากพอสมควร เพื่อใช้เป็นเนื้อหาในการทำความเข้าใจโครงงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# โครงสร้างข้อมูลพื้นฐาน

## 3.1 สแตก (Stack)

*นิยาม* สแตกเป็นรายชื่อแบบเชิงเส้นที่มีการแทรก (มักจะเรียกว่าการเพิ่มหรือพุช (push))และการนำออก (มักเรียกว่าการลบหรือป๊อป (pop)) เกิดขึ้นที่จุดปลายอันเดียวกัน จุดปลายนี้เรียกว่า ท๊อป (top) จุดปลายอีกด้านหนึ่งเรียกว่า บอตท้อม (bottom)




***รูปที่ 3-1 การทำงานของสแตก***

จากรูป 3-1 แสดงสแตกที่มี 4 องค์ประกอบ สมมติเราต้องการที่จะเพิ่มองค์ประกอบ E เข้าไปในสแตกของรูป (a) องค์ประกอบนี้จะต้องถูกวางบนสุดเหนือองค์ประกอบ D, ให้เราจึงได้เป็นดังรูป (b) ถ้าเราต้องการลบองค์ประกอบจากสแตกของรูป (b) มั้นจะต้องเป็นองค์ประกอบ E สิ่งที่ตามมาหลังจากการลบเป็นผลจากการปรับปรุงแก้ไขสแตกในรูป (a) ถ้าเราทำการลบ 3 ครั้งติดกันบนสแตกของรูป (b) สแตกของรูป (c) เป็นผลลัพธ์

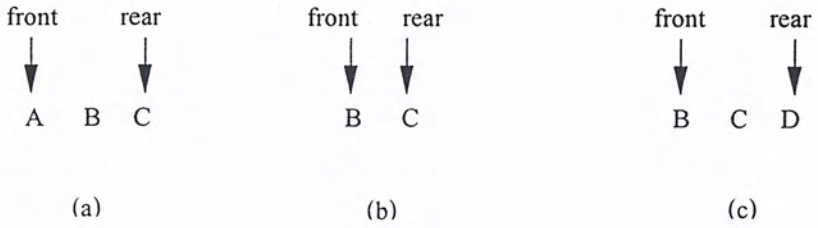
จากการแสดงข้างต้น, เราได้เห็นแล้วว่าสแตกเป็นลิสต์แบบ LIFO (Last-in-First-out เข้าทีหลัง-ออกก่อน) ลิสต์ของชนิดนี้จะปรากฏให้เป็นบ่อย ๆ ในการคิดของคอมพิวเตอร์

## 3.2 คิว (Queue)

คิวคล้ายกับสแตกเป็นลิสต์เชิงเส้นชนิดพิเศษ ในการแทรกและการลบของคิวทำในจุดปลายที่ต่างกันของลิสต์เชิงเส้น คิวเป็นลิสต์แบบ FIFO (first-in-first-out เข้าก่อน-ออกก่อน) คิวมีหลายชนิด เช่น คิวที่มีลำดับความสำคัญ (priority queue) เป็นต้น

*นิยาม* คิวเป็นลิสต์เชิงเส้นที่ซึ่งการแทรก (มักจะเรียกว่าการเพิ่มและใส่ (put)) และการลบ (มักจะเรียกว่าการนำออกมา (pop)) เกิดขึ้นที่จุดปลายต่างกัน จุดปลายที่ซึ่งองค์ประกอบอันใหม่ถูกเรียกว่า ท้าย (rear), และจากที่ที่องค์ประกอบอันเก่าถูกลบถูกเรียกว่า หน้า (front)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

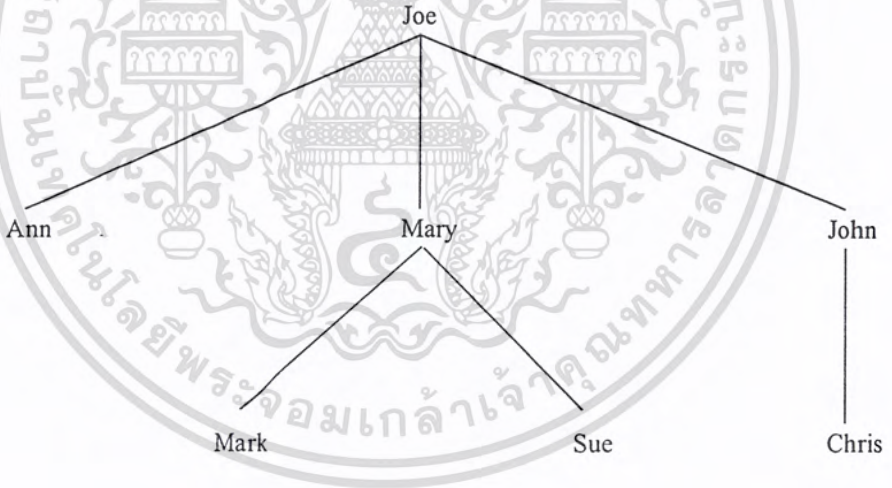



*รูปที่ 3-2 คิวอย่างง่าย*

คิวที่มี 3 องค์ประกอบถูกแสดงดังรูปที่ 3-2 องค์ประกอบแรกที่เราลบจากคิวของรูปที่ 3-2(a) คือ A สิ่งที่ตามมาหลังจากการลบ, การปรับปรุงแก้ไขรูปที่ 3-2 (b) เป็นผลลัพธ์ เพิ่มองค์ประกอบ D เข้าไปในคิวดังรูปที่ 3-2 (b), เราต้องวางมันตามหลัง C การแก้ไขครั้งใหม่ถูกแสดงดังรูปที่ 3-2 (c)

## 3.3 ทรี(Tree)

ในข้อมูลที่เป็นลำดับชั้นเราจะมีข้อมูลแม่ – ข้อมูลลูก, เหนือกว่า – ต่ำกว่า, ส่วนทั้งหมด, หรือคล้ายกับมีความสัมพันธ์ท่ามกลางส่วนย่อยของข้อมูล




***รูปที่ 3-3 ทายาทของโจ แสดงถึงโจและทายาทของเขาโดยจัดให้อยู่ในลำดับชั้น, เริ่มที่โจที่ชั้นบนสุดของลำดับชั้น ลูกของโจ (แอน, แมรี่, และจอห์น) ถูกจัดไว้ในลำดับถัดมาในลำดับชั้น, และเส้นโยงที่อยู่ร่วมกันในโจและลูกของเขา แอนไม่มีลูก, ขณะที่แมรี่มีลูก 2 คนและจอห์นมีลูก 1 คน ลูกของแมรี่ถูกวางอยู่ใต้เธอ, และลูกของจอห์นถูกวางอยู่ใต้เขา มีเส้นโยงระหว่างแต่ละพ่อแม่กับลูกของพวกเขา จากการแสดงลำดับชั้นนี้. มันง่ายที่จะบอกว่าใครเป็นพี่น้องของแอน, ใครเป็นทายาทของโจ, คริสมีบรรพบุรุษเป็นใคร และอื่น ๆ***

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*นิยาม* ทรี t เป็นกลุ่มขององค์ประกอบข้อมูลที่มีจุดจบและไม่ว่างเปล่า องค์ประกอบข้อมูลอันหนึ่งเรียกว่า ราก (root) ,และองค์ประกอบที่เหลือ (ถ้ามี) เป็นส่วนหนึ่งของทรีที่เรียกว่า ทรีย่อย (subtrees) ของ t

ลองมาดูว่านิยามนี้เกี่ยวข้องกับตัวอย่างของลำดับชั้นของเราอย่างไร องค์ประกอบที่ชั้นสูงสุดของลำดับชั้น คือ ราก องค์ประกอบที่ชั้นถัดมาเป็นรากของทรีย่อยที่อยู่ในรูปส่วนหนึ่งขององค์ประกอบที่เหลือ

จากตัวอย่างทายาทของโจที่ผ่านมา, กลุ่มข้อมูลเป็น {Joe, Ann, Mary, Mark, Sue, John, Chris} ดังนั้น n = 7 รากเป็นตัวเก็บข้อมูลของโจ องค์ประกอบที่เหลือถูกแบ่งเป็นส่วน 3 ส่วนแยกจากกัน {Ann}; {Mary, Mark, Sue}; และ {John, Chris} {Ann} เป็นทรีที่มีองค์ประกอบเพียงอันเดียว; รากของมันคือแอน รากของ{Mary, Mark, Sue} คือ แมรี่, และ {John, Chris} คือ จอห์น องค์ประกอบที่เหลือของ {Mary, Mark, Sue} ถูกแบ่งออกเป็นส่วนที่แยกจากกันเป็น {Mark} และ {Sue}, ที่ซึ่งทั้งคู่เป็นทรี(ย่อย)ที่มีเพียงองค์ประกอบอันเดียว, และองค์ประกอบที่เหลือของ {John, Chris} เป็นทรีย่อยที่มีองค์ประกอบเพียงอันเดียว

เมื่อวาดรูปทรีแล้ว, แต่ละองค์ประกอบของข้อมูลถูกแทนด้วยโหนด (node) รากของทรีถูกเขียนที่ชั้นบนสุด, และทรีย่อยของมันถูกเขียนลงมาข้างใต้ มีเส้นโยงจากรากของทรีไปยังรากของทรีย่อยของมัน (ถ้ามี) แต่ละทรีย่อยถูกเขียนคล้ายกับรากที่ชั้นบนสุดและทรีย่อยของมันข้างใต้ ที่ปลายในการเชื่อมต่อโหนดองค์ประกอบของทรีและโหนดลูกของมัน ในรูปที่ 3-3 ตัวอย่างเช่น แอน, แมรี่, และจอห์นเป็นลูกของโจ, และโจเป็นพ่อ-แม่ (parents) ของพวกเขา ลูกที่มีพ่อ-แม่คนเดียวกันถูกเรียกว่าพี่น้อง (siblings) แอน, แมรี่ ,และจอห์นเป็นพี่น้องกันในทรีรูปที่ 3-3 ,แต่มาร์คและคริสไม่ใช่พี่น้องกัน ส่วนขยายของเทอมต่อไปนี้รวมถึงเทอม หลาน (grandchild), ปู่-ย่า (grandparent), บรรพบุรุษ (ancestor), ทายาท (desendent) และถัดจากนี้ ในองค์ประกอบของทรีที่ไม่มีลูกจะเรียกว่าลีฟ (leaves) ดังนั้น แอน, มาร์ค, ซู, และคริส เป็นลีฟของทรีในรูปที่ 3-3 รากของทรีเป็นเพียงองค์ประกอบอันเดียวที่ไม่มีพ่อ-แม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง จุฬาลงกรณ์มหาวิทยาลัย



# บทที่ 4
# ทฤษฎีและการออกแบบระบบ

ในการทำงานโครงงานชิ้นนี้ เราจะมีการออกแบบในรูปแบบของการทำงานเชิงวัตถุ คือใช้ภาษาจาวาซึ่งถือเป็นภาษาที่ใช้แนวคิดเชิงวัตถุในการเขียน จึงขอกล่าวถึงเรื่องราวของการออกแบบโดยใช้ภาษายูเอ็มแอล (UML) ก่อนพอสังเขป

## 4.1 ยูเอ็มแอล (UML)

ยูเอ็มแอล (UML) หรือ Unified Modeling Language เป็นภาษาในการโมเดลมาตรฐาน ยูเอ็มแอลเกิดจากการพัฒนาร่วมกันของผู้นำเทคโนโลยีทางด้านวัตถุ 3 ท่านคือ แกรดี้ บูช (Grady Booch), อิวาร์ จาคอบสัน (Ivar Jacobson) และ จิม รัมบอช (Jim Rumbaugh) โดยก่อนที่จะมาเป็นยูเอ็มแอล นั้นผู้นำทั้ง 3 คนนี้ต่างก็มีโมเดลสำหรับการพัฒนาเชิงวัตถุเป็นของตัวเอง ต่อมาทางบริษัทเรชันแนลซอฟต์แวร์ (Rational Software) จึงได้ร่วมมือกับผู้นำทั้ง 3 คนนี้ในการพัฒนาโมเดลร่วมกัน เพื่อให้เกิดความเป็นหนึ่งเดียวสำหรับการพัฒนาระบบเชิงวัตถุ

ยูเอ็มแอลนั้นได้รวมแนวความคิดของวิธีการต่างๆ เข้าไว้ด้วยกัน จุดประสงค์ของยูเอ็มแอล ก็คือต้องการสร้างโมเดลในการพัฒนาที่เข้าใจและสร้างได้ง่าย แต่สามารถนำไปใช้ได้กับทุกๆระบบ ตัวยูเอ็มแอล เองนั้นได้รับการสนับสนุนจากผู้นำในอุตสาหกรรมทางด้านคอมพิวเตอร์หลายๆ ราย เช่น เอชพี (HP),ไอคอน คอมพิวติ้ง (ICON Computing) ,ไอบีเอ็ม (IBM) , ไอ-โลจิกซ์ (I-Logix), อินเทลลิคอรป์ (Intellicorp), เอ็มซีไอซิสเต็มเฮาส์ (MCI System house), ไมโครซอฟต์ (Microsoft), ออบเจกต์ทาม (ObjecTime), โอราเคิล (Oracle), แพลตตินัมเทคโนโลยีเพทค (Platinum Technology Ptech), ไรชเทคโนโลยี (Reich Technologies), ซอฟต์ทีม (Softeam), สเตอลิงซอฟต์แวร์ (Sterling Software), ทาสคอน (Taskon) และ ยูนิซิส (Unisys)

ในยูเอ็มแอลมีโมเดลที่สื่อสารด้วยภาพได้สำหรับระบบหลายๆ โมเดล โดยแต่ละโมเดลก็จะแสดงมุมมองต่อระบบที่ไม่เหมือนกัน ซึ่งประกอบไปด้วย

แผนภาพยูส-เคส (Use-Case) ที่ใช้แสดงการติดต่อระหว่างระบบกับผู้ใช้

แผนภาพคลาส ( Class Diagram ) ที่ใช้แสดงโครงสร้างทางตรรกของระบบ

แผนภาพวัตถุ (Object Diagram ) ใช้แสดงวัตถุและความสัมพันธ์ระหว่างวัตถุ

แผนภาพสถานะ(State Diagram ) ใช้แสดงพฤติการณ์ของระบบ

แผนภาพคอมโพเน็นต์ ( Component Diagram ) ใช้แสดงโครงสร้างทางกายภาพของซอฟร์แวร์

แผนภาพดีพลอยเมนต์ (Deployment) ที่ใช้แสดงการติดต่อระหว่างซอฟต์แวร์กับฮาร์ดแวร์

แผนภาพอินเตอแรคชัน (Interaction) ใช้แสดงพฤติการณ์ของระบบ

แผนภาพกิจกรรม (Activity Diagram) ใช้แสดงการไหลของอีเวนท์ (Event)ในยูสเคส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42785

ยูเอ็มแอลนั้นประกอบไปด้วยสัญลักษณ์และเครื่องหมายจำนวนมาก โดยสัญลักษณ์และเครื่องหมายจำนวนมากของยูเอ็มแอลเหล่านี้ ทำให้สามารถสร้างโมเดลแผนภาพได้อย่างยืดหยุ่น และครอบคลุมโดเมนของปัญหาทั้งหมด จะไม่ขอกล่าวถึงสัญลักษณ์ทั้งหมดของ ยูเอ็มแอล แต่ละกล่าวถึงสัญลักษณ์บางอย่างที่จำเป็นต้องใช้ได้แก่

- เท็กซ์โน้ต (Text Note)
- คอนสเตรน (Contrain)
- สเตริโอไทป์ (Stereotype)

**4.1.1 เท็กซ์โน้ต (Text Note)**

เป็นส่วนหนึ่งของแผนภาพที่ไม่มีความหมายต่อโมเดลแต่อย่างใด โดยทั่วไปจะใช้เท็กซ์โน้ตเพื่ออธิบายสิ่งที่อยู่ในแผนภาพบางสิ่งเพื่อให้ผู้ที่อ่านหรือใช้ไดอะแกรมเข้าใจความหมายได้ดีขึ้น โดยปกติเท็กซ์โน้ต จะเป็นตัวอักษรที่อยู่มุมบนขวาขององค์ประกอบ ในแผนภาพ

**4.1.2 คอนสเตรน (Constrain)**

เป็นเงื่อนไขที่เพิ่มเข้ามาสำหรับองค์ประกอบในแผนภาพ ตัวอย่างเช่น ในเงื่อนไขของอินพุตที่อยู่ในแผนภาพสถานะ โดยเงื่อนไขอินพุตนี้ จะเป็นข้อกำหนดที่ใช้แสดงว่าระบบจะต้องตอบสนองต่ออินพุตที่มีเข้ามาจากภายนอก โดยปกติคอนสเตรน จะมีอยู่ในแยกหรือเคอร์ลี บราซส์ (Curly Braces) และอาจจะปรากฏอยู่ในเท็กซ์แอเรีย (Text Area) ก็ได้

**4.1.3 สเตริโอไทป์ (Stereotype)**

เป็นเมต้าโมเดล (Metamodel) ของยูเอ็มแอล หรือจะกล่าวง่ายๆว่าสเตริโอไทป์เป็นเอ็นติตี้ (Entity) ซึ่งเป็นส่วนขยายสำหรับยูเอ็มแอล เพราะถึงแม้ว่าในยูเอ็มแอลนั้นจะมีสัญลักษณ์และเครื่องหมายจำนวนมาก แต่ก็ยังไม่เพียงพอที่จะใช้แทนทุกๆสิ่งที่มีในโลกความเป็นจริง แต่สามารถใช้เมต้าโมเดลนี้ในการแทนความหมายของสิ่งเหล่านตามที่ต้องการได้

อย่างไรก็ดี สัญลักษณ์และเครื่องหมายของยูเอ็มแอลทั้งหมดสามารถเขียนอยู่ในรูปของ สเตริโอ-ไทป์ (Stereotype) ได้เช่นกัน ตัวอย่างเช่น ในรูปสามารถเขียนเมสเซจที่มี ลักษณะแบบ เอ็พพีโซดิก-เวตติ้ง (Episodic-Waiting) ในยูเอ็มแอล ได้ตามรูป A และสามารถเขียนเป็นสเตริโอได้ในรูป B นอกจากนั้นยังสามารถสร้างเมต้าคลาส (Metaclass) โดยขยายข้อกำหนดหรือ เพิ่มความเฉพาะเจาะจงให้กับสัญลักษณ์หรือเครื่องหมายของยูเอ็มแอลที่มีอยู่ได้ ซึ่งคุณสมบัตินี้ก็เหมือนกันการสืบทอดคุณสมบัตินั่นเอง

สำหรับสัญลักษณ์ของสเตริโอไทป์ (Stereotype) นั้นจะเขียนชื่อเอ็นติตี้ (Entity) ที่จะเป็น สเตริโอไทป์ภายในวงเล็บ <<>> เช่น "<<type>> tree " เป็นการกำหนดสเตริโอไทป์ของชนิดข้อมูลเพื่อใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนโปรแกรมเมื่อผู้ดูไดอะแกรมเห็นสัญลักษณ์ของสเตริโอไทป์นี้จะเข้าใจทันทีว่าข้อมูลนั้นมีชนิดเป็นทรีตัวยูเอ็มแอลก็มีการกำหนดสัญลักษณ์ของ สเตริโอไทป์ พื้นฐานไว้สำหรับแอพพลิเคชันเฉพาะด้านเช่นกัน

เอ็นติตี้ในโมเดลของยูเอ็มแอลทุกตัวสามารถเขียนเป็นสเตริโอไทป์ได้ ตัวอย่างเช่น เมสเซจบางตัวอย่างของสเตริโอไทป์

**4.2 รูปแบบของการออกแบบ ( Design Pattern )**

ดีไซน์แพตเทิร์น นั้นเป็นการนำรูปแบบหรือแนวความคิดในการออกแบบระบบ กลับมาใช้ใหม่ นอกจากนั้นดีไซน์แพตเทิร์นยังช่วยปรับปรุงการทำเอกสารประกอบการออกแบบรวมทั้งช่วยในการดูแลระบบที่มีอยู่

สรุปง่ายๆ ก็คือการใช้ดีไซน์แพตเทิร์นนั้นทำให้ผู้ออกแบบระบบสามารถออกแบบระบบที่ถูกต้องได้โดยใช้เวลาน้อยลง

โดยทั่วไปแพตเทิร์นจะประกอบด้วยส่วนหลัก 4 ส่วน คือ

- ชื่อของแพตเทิร์น

ชื่อของแพตเทิร์นจะบ่งบอกถึงปัญหาของการออกแบบ ทางออกสำหรับปัญหาและอาจจะเป็นคำที่บ่งบอกถึงประเด็นหลักของแพตเทิร์นนั้น

- ปัญหา

ปัญหาเป็นส่วนที่อธิบายว่าปัญหาและสิ่งที่เกี่ยวข้องกับระบบ ในส่วนของปัญหาอาจจะอธิบายปัญหาในการออกแบบเฉพาะตัวอย่างเช่นอัลกอริทึมที่ใช้ในวัตถุ หรืออาจจะอธิบายโครงสร้างของคลาสและวัตถุ

ในบางครั้งปัญหาอาจจะระบุเงื่อนไขของระบบที่จะนำแพตเทิร์นนั้นๆไปประยุกต์ใช้ด้วย

- โซลูชัน

โซลูชันจะอธิบายส่วนประกอบที่ต้องมีในการออกแบ ความสัมพันธ์ความรับผิดชอบและการทำงานร่วมกัน ในส่วนโซลูชันจะไม่ได้อธิบายรูปแบบการออกแบบหรือการพัฒนา

โดยเฉพาะเจาะจงเนื่องจากว่าดีไซน์แพตเทิร์นนั้นสามารถนำไปประยุต์ใช้กับระบบทั่วไป แต่ตัวดีไซน์แพตเทิร์นจออธิบายถึงปัญหาในการออกแบบและวิธีการวางโครงสร้าการทำงานของคลาสและวัตถุเพื่อแก้ปัญหานั้นๆ

- บทสรุป

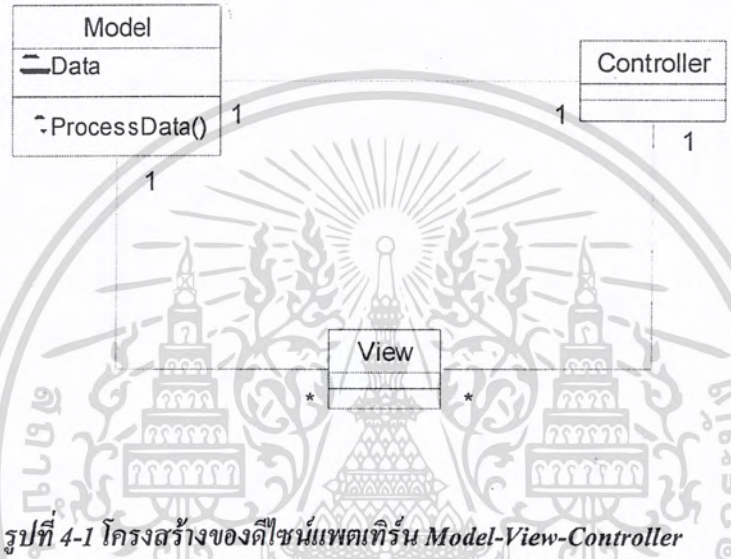
บทสรุปจะอธิบายถึงข้อดีข้อเสียในการประยุกต์ใช้ดีไซน์แพตเทิร์น บางครั้งผู้ออกแบบระบบควรจะศึกษาบทสรุปของดีไซน์แพตเทิร์นก่อนที่จะนำดีไซน์แพตเทิร์นนั้นไปประยุกต์ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งนี้เนื่องจากปัจจัยหลักในการออกแบบระบบด้วยวิธีเชิงวัตถุคือการนำกลับไปใช้ใหม่ ดังนั้น บทสรุปของดีไซน์แพตเทิร์นจะอธิบายความยืดหยุ่น ความสามารถในการขยายหรือความสามารถในการนำไปใช้กับระบบใหม่หลังจากที่ได้มีการประยุกต์ใช้ดีไซน์แพตเทิร์นเข้าไปแล้ว

## 4.3 ดีไซน์แพตเทิร์นที่มีการนำมาใช้ในระบบ

แพตเทิร์นโมเดล-วิว-คอนโทรลเลอร์ (Model-View-Controller)




*รูปที่ 4-1 โครงสร้างของดีไซน์แพตเทิร์น Model-View-Controller*

จุดประสงค์

แพตเทิร์นโมเดล-วิว-คอนโทรลเลอร์ (Model-View-Controller) เป็นแพตเทิร์นที่ช่วยในการแยกส่วนในการตอบสนองกับผู้ใช้ ส่วนแสดงผลและส่วนประมวลผลข้อมูลออกจากกัน ทั้งนี้เพื่อให้ผู้ออกแบบระบบสามารถนำส่วนประกอบในแพตเทิร์นกลับไปใช้ใหม่และยังทำให้ระบบมีความยืดหยุ่นต่อการเปลี่ยนแปลงในส่วนใดส่วนหนึ่งด้วย

การประยุกต์ใช้

ถ้าในระบบได้รับการกำหนดให้วัตถุหนึ่งทำหน้าที่ทั้งติดต่อกับผู้ใช้ ประมวลผลข้อมูลและแสดงผลแล้วผู้ออกแบบระบบสามารถแยกวัตถุนั้นออกเป็น 3 วัตถุได้ที่ประกอบไปด้วย

- โมเดลซึ่งเป็นส่วนประมวลผลข้อมูล
- วิวทำหน้าที่แสดงผลข้อมูล
- คอนโทรลเลอร์ทำหน้าที่รับและประมวลผลอีเวนท์ที่มีเข้ามาจากวัตถุวิว

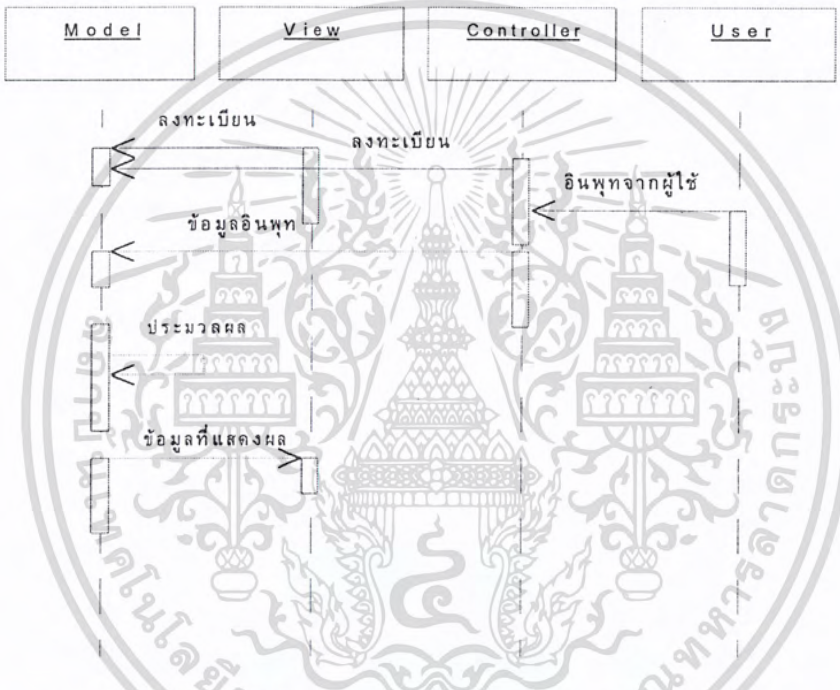
ผู้ออกแบบระบบสามารถประยุกต์ใช้ดีไซน์แพตเทิร์นโมเดล-วิว-คอนโทรลเลอร์ นี้กับแพตเทิร์นอื่นได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>บทบาทของแต่ละส่วนประกอบในดีไซน์แพตเทิร์น</u>

- โมเดลเป็นวัตถุที่เก็บและประมวลผลข้อมูลที่สำคัญในระบบ วัตถุโมเดลจะรับข้อมูลจากวัตถุคอนโทรลเลอร์ และแสดงผลข้อมูลออกที่วัตถุวิว โดยทั่วไปวัตถุโมเดลจะเป็นวัตถุหลักของระบบ
- วิวเป็นวัตถุที่รับข้อมูลจากวัตถุโมเดลเพื่อแสดงผล
- คอนโทรลเลอร์เป็นวัตถุที่ทำหน้าที่ติดต่อหรือรับอินพุตจากผู้ใช้และส่งไปให้กับวัตถุโมเดลเพื่อประมวลผล

<u>โครงสร้างการทำงาน</u>




*รูปที่ 4-2 โครงสร้างการทำงานของดีไซน์แพตเทิร์นโมเดล-วิว-คอนโทรลเลอร์*

1. วัตถุวิวและคอนโทรลเลอร์ลงทะเบียนกับวัตถุโมเดล เพื่อให้วัตถุโมเดล ทราบว่ามีวัตถุใดที่ทำหน้าที่รับอินพุตและแสดงผลต่อผู้ใช้
2. เมื่อมีอินพุตมาจากผู้ใช้วัตถุคอนโทรลเลอร์จะทำหน้าที่ส่งข้อมูลนั้นไปให้กับวัตถุโมเดล
3. วัตถุโมเดล เมื่อได้รับข้อมูลอินพุตแล้วก็จะประมวลผลซึ่งในกรณีนี้อาจจะใช้ข้อมูลประกอบการประมวลผลอื่นๆ ด้วย
4. วัตถุโมเดลส่งข้อมูลที่ต้องการแสดงผลไปให้กับวัตถุวิว เพื่อจัดการแสดงผลต่อผู้ใช้

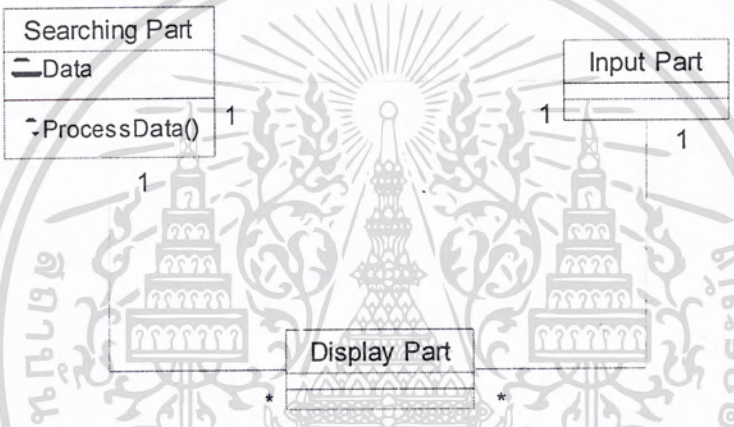
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ผลลัพธ์ที่ได้</u>

ทำให้มีการแยกแยะส่วนประมวลผล ส่วนแสดงผลและส่วนรับอินพุตออกจากกันทำให้ระบบมีความยืดหยุ่น ในการปรับเปลี่ยนในส่วนใดส่วนหนึ่งเพราะจะไม่กระทบกับส่วนอื่นนอกจากนั้นยังสนับสนุนการนำกลับไปใช้ใหม่ของวัตถุในแพตเทิร์นด้วยตัวอย่างเช่นผู้ออกแบบสามารถนำวัตถุวิว ไปใช้กับระบบอื่นได้ถ้าระบบใหม่นั้นมีส่วนแสดงผลเหมือนกัน

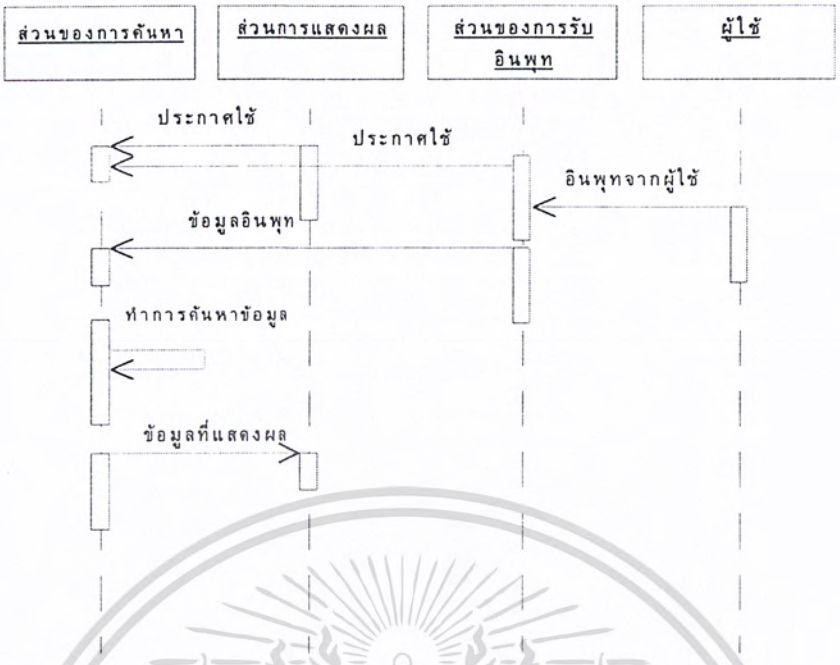
## 4.4 การนำมาประยุกต์ใช้ในการออกแบบ

เมื่อเรานำเอาดีไซน์แพตเทิร์น ดังกล่าวมาประยุกต์ใช้กับการออกแบบโครงงาน โดยเราจะมองโครงงานออกเป็น 3 ส่วน คือ 1 ส่วนของการประมวลผล 2 ส่วนของการอินพุตข้อมูล และ 3 ส่วนของการแสดงผลข้อมูล โดยเราจะได้ลักษณะเป็นดังนี้




*รูปที่ 4-3 โครงสร้างของดีไซน์แพตเทิร์นของโครงงาน*

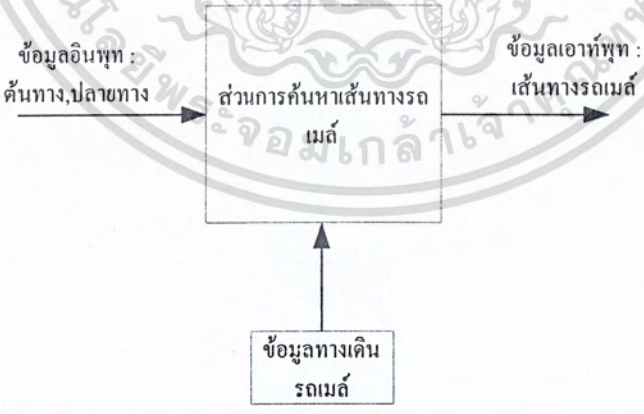
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 4-4 โครงสร้างของดีไซน์แพตเทิร์นของโครงงาน*

**4.5 การออกแบบส่วนการค้นหา**

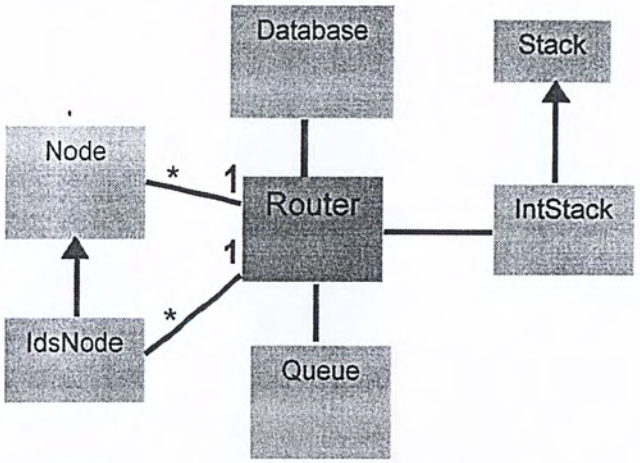
ส่วนนี้เป็นการหาเส้นทางรถเมล์โดยจะนำหลักการของ โครงสร้างข้อมูล และปัญญาประดิษฐ์มาใช้ มีจุดมุ่งหมายที่ต้องการต่อรถเมล์น้อยที่สุด เขียนโดยภาษาจาวา




*รูปที่ 4-5 แสดงการติดต่อข้อมูลในส่วนของการค้นหาเส้นทางรถเมล์*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ข้อมูลความต้องการของระบบแล้วก็จะทำการหาออบเจ็กต์ ได้ออบเจ็กต์ดังรูปที่ 4-6




*รูปที่ 4-6 แสดงออบเจ็กต์ และการติดต่อระหว่างกัน*

1. เราต์เตอร์ (Router) เป็นวัตถุที่ใช้หาเส้นทาง โดยจะรับข้อมูลอินพุต (ต้นทาง,ปลายทาง,วิธีการค้นหา) มาจากส่วนการติดต่อกับผู้ใช้ โดยมีเมธอดที่สำคัญอยู่ 3 อัน คือ

- breadth method เป็นการค้นหาแบบเบรดท์-เฟิรสท์ เซิรสช์
- iterative method เป็นการค้นหาแบบอิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์
- findwayback method เป็นการหาและจัดเรียงเส้นทาง

2. คาต้าเบส (Database) เป็นวัตถุที่ใช้เก็บข้อมูลสายรถเมล์และป้ายรถเมล์ ให้อยู่ในรูปแบบที่ง่ายต่อการนำไปใช้ โดยมีแอดทริบิว (attribute) ที่สำคัญอยู่ 2 อัน คือ

- BusLine [ ] เป็นการเก็บข้อมูลว่า รถเมล์สายต่าง ๆ วิ่งผ่าน ป้ายรถเมล์ใดบ้าง ตัวอย่างเช่น

| สายรถเมล์ | ป้ายรถเมล์ |
|---|---|
| 1 | 121,23,45,55 |
| 2 | 111,85,23,55 |

- BusStop [ ] เป็นการเก็บข้อมูลว่า ป้ายรถเมล์นั้นมีรถเมล์สายใดวิ่งผ่าน ตัวอย่างเช่น

| ป้ายรถเมล์ | สายรถเมล์ |
|---|---|
| 121 | 1,3,45 |
| 23 | 1,2,3,50 |

3. โหนด (Node) เป็นตัวเก็บข้อมูลโหนดในทรี ที่ใช้ในการค้นหาแบบเบรดท์-เฟิรสท์ เซิรสช์ มีแอดทริบิวอยู่ 4 ตัว คือ

- busline ชื่อสายรถเมล์(ชื่อโหนด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- startfrom ขึ้นรถเมล์จากป้ายใด
- spanfrom เป็นโหนดลูกของโหนดใด
- isgoal เป็นโกลหรือไม่

4. คิว (Queue) เป็นโครงสร้างข้อมูลที่ใช้เก็บลำดับของโหนดในทรี ที่ใช้ในการค้นหาแบบเบรดท์-เฟิรสท์ เซิรสช์ มีเมธอดสำคัญอยู่ 2 ตัว คือ

- enqueue เก็บค่าใส่คิว
- dequeue นำค่าในคิวออกมา

5. สแตก (Stack) เป็นโครงสร้างข้อมูลที่ใช้เพื่อเก็บลำดับแบบเข้าก่อน-ออกทีหลัง

6 อินท์สแตก (Intstack) เป็นสแตกที่เก็บข้อมูลแบบอินทีเจอร์ (integer) โดยมีเมธอดสำคัญอยู่ 2 ตัว คือ

- Intpush ใส่ข้อมูลลงในสแตก
- Intpop นำข้อมูลออกจากสแตก

7. อินท์โหนด (Intnode) เป็นโหนดที่ใช้ในการหาแบบอิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์

นำส่วนการค้นหาของปัญญาประดิษฐ์มาใช้ โดยในโครงงานชิ้นนี้ได้ใช้

1. อัลกอริทึมแบบ เบรดท์-เฟิรสท์ เซิรสช์ (เบรดท์-เฟิรสท์ เซิรสช์) เพราะอัลกอริทึมนี้เป็นการหาทางที่สั้นที่สุด (Optimal path) และครบทุกเส้นทาง (Completeness)
2. อัลกอริทึมแบบ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ (Iterative Deepening search) เพราะอัลกอริทึมนี้เพราะได้รวมข้อดีของเบรดท์-เฟิรสท์ เซิรสช์ ที่สามารถหาทางที่สั้นที่สุด และครบทุกเส้นทาง กับ เดพท์-เฟิรสท์ เซิรสช์ ที่ใช้หน่วยความจำน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# อัลกอริทึมการค้นหา

## 5.1 อัลกอริทึมการค้นหา

อัลกอริทึมเป็นชุดคำสั่งที่มีความแน่นอนสำหรับการทำให้บรรลุถึงจุดมุ่งหมาย ในโปรแกรมคอมพิวเตอร์, โครงสร้างข้อมูลทั้งหมดมีส่วมร่วมอยู่ในอัลกอริทึมที่ถูกใช้เข้าถึงและแก้ไขข้อมูลที่มันเก็บไว้อยู่ โครงสร้างข้อมูลของจาวาที่สร้างมาให้อยู่แล้ว เช่น int และ float ,มีอัลกอริทึมอยู่ข้างใน เช่น การคำนวณทางตัวเลข (arithmetic) และ การเปรียบเทียบ (comparison) คลาสที่ผู้ใช้สร้างขึ้นมาเองก็จะมีอัลกอริทึมซ่อนเก็บไว้ในเมธอดของคลาสเหล่านั้น

สำหรับการปฏิบัติการมากมายทำงานบนโครงสร้างข้อมูล, มีหลาย ๆ อัลกอริทึมที่แตกกต่างกันอาจจะถูกใช้ อัลกอริทึมที่แตกต่างกันทำให้เกิดความต้องการที่แตกต่างกันบนหน่วยความจำของคอมพิวเตอร์และเวลาในการทำงาน ในการเซตที่มาตั้งแต่การผลิต ที่ซึ่งราคาของซอฟต์แวร์เพื่อการค้าและเวลาของพนักงานเป็นสิ่งสำคัญ ด้วยความหาได้ง่ายและราคาของงานที่ไม่ใช่งานประจำที่ต้องพัฒนาอัลกอริทึม และเวลาที่ใช้ในการโปรแกรมเพื่อพัฒนาอัลกอริทึม ถ้าไม่มีผลิตภัณฑ์วางขายอยู่ ต้องนำมาพิจารณา

อย่างไรก็ตาม ประโยชน์ที่รู้ว่าบางสิ่ง เช่น เวลาโปรแกรมทำงานและความต้องการของหน่วยความจำของอัลกอริทึม ดังนั้นคุณสามารถทำการตัดสินใจอย่างเปิดเผยเกี่ยวกับอัลกอริทึมที่ดีที่สุดสำหรับงานชิ้นนั้นของคุณ การศึกษาทางทฤษฎีของอัลกอริทึม รู้จักกันในชื่อ ทฤษฎีความซับซ้อน (Complexity Theory) เป็นสาขาที่ใช้คณิตศาสตร์ชั้นสูงสาขาหนึ่งของวิทยาศาสตร์คอมพิวเตอร์ มันเกี่ยวกับการวิเคราะห์แบบละเอียดของจำนวนขั้นตอนการทำงาน (step) ที่ต้องการโดยอัลกอริทึมเพื่อประมวลผลข้อมูลต่าง ๆ สำหรับหลาย ๆ อัลกอริทึม, ผลลัพธ์จริงถูกส่งออกมา, แต่สำหรับอัลกอริทึมแบบอื่น ทำเพียงแค่ประมาณหรือเฉลี่ยค่าออกมา

### 5.1.1 ประสิทธิภาพของอัลกอริทึม

<u>การจัดแบ่งอัลกอริทึมโดยประสิทธภาพของมันเอง</u>

จากที่ได้เห็นอาจจะมี 2 อัลกอริทึมที่ได้คำตอบมาเหมือนกัน และก็มีประสิทธิภาพที่แตกต่างกัน และมีทางที่เป็นระบบของการหาและการแบ่งประสิทธิภาพ

มีการเข้าถึงการศึกษาของอัลกอริทึม, แต่พวกเราไม่ได้ถูกสมมติว่ามีความชำนาญทางคณิตศาสตร์มากมายในส่วนของผู้อ่าน เรามักจะติดกับกับวิธีหาทางที่ดีที่สุดในขณะนั้น (heuristic)และหาทางแบบทดลอง (experiment) การจัดเรียงแบบปกติของตัวอากิวเมนต์และเมธอดที่เราจะใช้ที่เราให้อากิวเมนต์ง่าย ๆ ในการประมาณประสิทธิภาพอัลกอริทึมการค้นหา สำหรับบางอัลกอริทึม เช่น อากิวเมนต์ค่อนข้างง่ายที่จะสร้าง สำหรับตัวอื่น ๆ, ดูยากกว่าในการพิจารณา สิ่งหนึ่งที่เราสามารถทำได้เสมอ ,แม้จะเป็นลำดับของการทดลองคอมพิวเตอร์ที่ซึ่งเรานับจำนวนขั้นตอนการทำงานที่ใช้ในการรันโปรแกรมบนชุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดต่าง ๆ กันไป เราใช้วิธีนี้ และจัดการการสร้างบางพร็อตของจำนวนขั้นตอนการทำงาน เช่น ฟังก์ชันของความยาวของรายชื่อที่กำลังค้นหาอยู่

เราพบว่า, ทั้งวิธีอากิวเมนต์หาทางที่ดีที่สุดขณะนั้นและหาทางแบบทดลอง และวิธีการทดลองคอมพิวเตอร์, ซึ่งจำนวนเฉลี่ยของการเปรียบเทียบที่ต้องการในการใช้การค้นหาแบบต่อเนื่อง (sequential search) เป็นสัดส่วนโดยตรงกับความยาวของรายชื่อ (list) ที่จะค้นหา การใช้วิธีเดียวกันนี้ เราพบว่าที่จำนวนเฉลี่ยของการเปรียบเทียบที่ต้องการในการค้นหาแบบฐาน 2 (binary search) เป็นสัดส่วนกับกำลังของ 2 ที่ต้องถูกเพิ่มเพื่อให้ได้ความยาวของรายชื่อ (นั่นคือ ถ้าความยาวรายชื่อคือ $2^n$ , จำนวนของการเปรียบเทียบเป็นสัดส่วนกับ n)

เราได้พบว่า ทั้งอากิวเมนต์ทางทฤษฎีหรือการทดลองกับคอมพิวเตอร์ พบว่าจำนวนขั้นตอนการทำงานที่ใช้โดยอัลกอริทึมขึ้นอยู่กับบางเงื่อนไขที่เกี่ยวกับจำนวนข้อมูลที่กำลังประมวลผลอยู่ ตัวอย่างเช่น ในกรณีของการค้นหาแบบต่อเนื่อง เราสามารถบอกจำนวน C ของการเปรียบเทียบว่าเท่ากับครึ่งหนึ่งของความยาว L ของรายชื่อ สามารถถูกเขียนได้เป็น $L = 2^n$, แล้วจำนวนของการเปรียบเทียบจะประมาณได้เท่ากับ n: $C = n$

สิ่งที่สำคัญที่สุดในการสังเกตเกี่ยวกับประสิทธิภาพของการค้นหาแบบต่อเนื่องว่า C เป็น สัดส่วนโดยตรงกับ L ในความจริงมีปัจจัยสเกลาร์คือ ½ แม้ว่าประโยชน์เป็นที่ควรจะทราบ คือ ไม่ใช่บิตส่วนสำคัญที่สุดของข้อมูล การทราบว่า C เป็นสัดส่วนตรงกับ L หมายถึงว่าเราทราบว่าถ้าความยาวของรายชื่อเป็นสองเท่า, จำนวนการเปรียบเทียบที่ใช้จะเพิ่มเป็น 2 เท่าด้วย ถ้าความยาวเพิ่มเป็น 3 เท่า, จำนวนการเปรียบเทียบก็เพิ่มเป็น 3 เท่าด้วย ในทางกลับกัน อะไรเป็นส่วนสำคัญในประโยค C เป็นสัดส่วนโดยตรงกับ L , และไม่ใช่อะไรมีค่าคงที่ของสัดส่วนอยู่ ในจุดหลักของการหาประสิทธิภาพของอัลกอริทึม คือ รู้ว่ามันประพฤติตัวอย่างไรสำหรับจำนวนข้อมูลขนาดใหญ่ ในส่วนย่อย, งานเร็วเพิ่มขึ้นหรือลดลงเพียงใด

เราสามารถสร้างอากิวเมนต์สำหรับแบ่งเป็น 2 อย่างในทุกอัลกอริทึมในจำนวนขั้นตอนเพิ่มในสัดส่วนโดยตรงกับจำนวนข้อมูล อย่างเช่นอัลกอริทึมที่เรียกว่า อัลกอริทึมเชิงเส้น (linear algorithms)

ในทางเดียวกัน เราสามารถแบ่งอัลกอริทึมออกเป็น 2 แบบทุกอัลกอริทึมที่ทำตัวเหมือนกับการค้นหาแบบฐาน 2 การเรียงลำดับของอัลกอริทึมถูกเรียกว่า อัลกอริทึมแบบลอกการิทึม (logarithmic algorithms) ถ้าคุณรู้ว่าลอกการิทึมถูกให้ความหมายอย่างไรในคณิตศาสตร์, คุณจะเห็นว่าทำไมสิ่งนี้ถึงเป็นจริง ถ้าคุณไม่ทราบว่าลอกการิทึมคืออะไร คำอธิบายแบบสั้น ๆ อยู่ในลำดับถัดไป

พิจารณาถึงเงื่อนไขอันหนึ่งที่พวกเราใช้ในก่อนหน้านี้สำหรับความยาวของรายชื่อในเทอมของกำลัง 2 : $L = 2^n$ ลอกการิทึมของ L (ในลักษณะฐาน 2) ถูกบอกลักษณะให้เป็น n นั่นคือ $\log L = n$, ถ้า $L = 2^n$ ในการใช้คำ ลอกการิทึม(ฐาน 2) ของจำนวนของกำลังที่ 2 ต้องถูกเพิ่มจำนวน ลอกการิทึมของ 2 คือ 1 (เพราะ $2^1 = 2$) ลอกการิทึมของ 4 คือ 2 (เพราะ $2^2 = 4$) ลอกการิทึมของ 8 คือ 3 (เพราะ $2^3 = 8$) และไปเรื่อย ๆ

ลอกการิทึมของจำนวนที่ไม่เป็นกำลัง 2 สามารถถูกอธิบายการใช้เทคนิคทางคณิตศาสตร์ ยกตัวอย่างเช่น เราสามารถอธิบายลอกการิทึมของ 5 ให้เป็นกำลังที่ซึ่ง 2 ต้องถูกเพิ่มขึ้นเป็น 5 ตั้งแต่ที่ 5 ไม่ใช่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำลัง 2 ที่แท้จริง, แต่อยู่ระหว่าง 4 และ 8 คุณควรคาดว่าลอกการิทึมของ 5 อยู่ในช่วงระหว่าง 2 และ 3, และคุณอาจจะทายถูก (มันคือ 2.322 ถึง 3 ในการวางฐาน 10)

อัลกอริทึมการค้นหาแบบฐาน 2 มีประสิทธิภาพที่สามารถถูกเขียนได้ดังนี้ C = log L ที่ซึ่ง C เป็นจำนวนการเปรียบเทียบ และ L เป็นความยาวของรายชื่อ

### 5.1.2 บิ๊ก-โอ โนเตชัน (THE BIG-OH NOTATION)

ครั้งหนึ่งคุณเคยมีความคิดเกี่ยวกับการแยกประเภทอัลกอริทึมโดยทางปกติ พวกมันขึ้นอยู่กับจำนวนของข้อมูล คุณจะพบการประกาศจดสั้น (shorthand notation) เพื่อเพิ่มประสิทธิภาพที่ค่อนข้างจะนำมาจัดการได้ง่าย เราสามารถยืมการประกาศจากคณิตศาสตร์ เราสามารถนำเสนออัลกอริทึมเชิงเส้น (ที่ซึ่งจำนวนขั้นตอนการทำงานเพิ่มขึ้นในสัดส่วนโดยตรงกับจำนวนข้อมูล) โดยการประกาศ O(n), ที่ถูกอ่านว่า 'order *n*' การประกาศนี้หมายความว่า, สำหรับจำนวนของ n (ที่ n เป็นปริมาณที่วัดจากจำนวนข้อมูลที่ถูกประมวลผลโดยอัลกอริทึม n ที่เท่ากับ L เมื่อเรากำลังพูดถึงความยาวของรายชื่อข้างต้น เป็นต้น) จำนวนของขั้นตอนการทำงานที่ถูกนำโดยอัลกอริทึมเป็นสัดส่วนโดยตรงกับจำนวนข้อมูลที่กำลังถูกประมวลผล

ข้อจำกัดของคำนิยามนี้ 'ค่า n ที่สูง' ถูกยอมรับเพราะว่าในบางกรณีฟอร์มที่แท้จริงของการยึดติดกับจำนวนขั้นตอนการทำงานบนจำนวนข้อมูลที่สมนัยโดยการใช้ O(n) ไม่กลายมาเป็นความแน่นอนจนกระทั่ง n มีค่ามาก ตัวอย่างเช่น เรากล่าวถึงข้างต้นว่าจำนวนเฉลี่ยของการเปรียบเทียบที่ใช้ในการค้นหาแบบต่อเนื่องบนรายชื่อของความยาว L คือ (L+1)/2, ซึ่งขยายมาจาก L/2 + ½ เทอมพิเศษของ ½ หมายถึงว่าจำนวนของการเปรียบเทียบไม่ได้เป็นสัดส่วนคงที่กับ L อย่างไรก็ตาม, เมื่อรายชื่อที่ได้รับยาวมาก ดังนั้น L จะมีค่ามาก, ½ นี้ช่วยลดความสำคัญและบอกว่าจำนวนการทำงานเป็นสัดส่วนกับ L ให้มีความถูกต้องมากขึ้นเรื่อย ๆ การใช้ O(n) หมายความว่า เทอมที่เพิ่มขึ้นอย่างเร็วที่สุดเป็นจำนวนสัดส่วนหนึ่งของ n มีเทอมลำดับที่ต่ำลงมาอื่น (นั้นคือ เทอมที่ไม่เปลี่ยนเร็วเท่ากับที่ n เพิ่ม) แต่นั่นจะกลายเป็นส่วนที่ไม่สำคัญเหมือนกับ n ที่มีค่ามาก

ในกรณีของการค้นหาแบบฐาน 2 เทอมที่มีลำดับสูงสุด (เทอมที่เพิ่มเร็วที่สุด) เป็นเทอมของลอกการิทึม, ดังนั้นเราพูดว่าการค้นหาแบบฐาน 2 เป็นอัลกอริทึม O(log n)

ดังนั้น, ทุกอัลกอริทึมสามารถถูกแบ่งประเภทโดยการให้ลำดับของมัน, ในรูปแบบของการประกาศ บิ๊ก-โอ ในการทำเช่นนี้ แน่นอนเราต้องหาเงื่อนไขอันดับแรกสำหรับการยึดติดกับจำนวนขั้นตอนการทำงานในอัลกอริทึมบนจำนวนข้อมูลที่กำลังถูกประมวลผล, โดยบางรูปแบบของอากิวเมนต์แบบหาทางที่ดีที่สุดในขณะนั้น, หรือโดยการทดลองกับคอมพิวเตอร์ อย่างไรก็ตามเราได้ทำมัน เราถูกสนใจเป็นหลักในเทอมของเงื่อนไขที่เพิ่มขึ้นเร็วที่สุด เช่นเดียวกับจำนวนข้อมูลที่เพิ่มขึ้น

ลำดับพื้นฐานส่วนใหญ่ของอัลกอริทึม คือ สิ่งที่ตามมา, รายชื่อในลำดับการเพิ่มขึ้นของความซับซ้อน

- ค่าคงที่ (O(1)) จำนวนขั้นตอนการทำงานไม่ขึ้นอยู่กับจำนวนของข้อมูล ผลของกรณีที่ดีที่สุดสำหรับการค้นหาแบบต่อเนื่องคือ O(1) ,ถ้ามีส่วนที่ไว้ให้เราสำหรับการหาส่วนแรกของรายชื่อ, การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปรียบเทียบเพียง 1 ครั้งถูกมาใช้ไม่ว่ารายชื่อจะยาวเท่าไร จำไว้ว่า แม้ว่า O(1) หมายความว่า จำนวนคงที่ของขั้นตอนการทำงานที่ใช้, ทีซึ่งจำนวนคงที่สามารถเป็นจำนวนใด ๆ ก็ได้ ไม่จำเป็นต้องเป็นตัวใดตัวหนึ่ง อัลกอริทึมต้องการ 15 ขั้นตอนเสมอ ก็ยังคงเป็นอัลกอริทึม O(1)

- ลอกการิทึม (O(logn)) อัลกอริทึมเหมือนการค้นหาแบบฐานสอง

- เชิงเส้น (O(n)) อัลกอริทึมเหมือนกรณีเฉลี่ยของการค้นหาแบบต่อเนื่อง

- ลอก-เชิงเส้น (O(n log n)) อัลกอริทึมที่เทอมต้นเป็นสัดส่วนกับผลลัพธ์ของจำนวนข้อมูล n โดยลอกการิทึมของจำนวนข้อมูล (log n) อัลกอริทึมการเรียงลำดับที่มีประสิทธิภาพส่วนใหญ่เป็นชนิดนี้,

- ยกกำลัง 2 ($O(n^2)$) อัลกอริทึมที่จำนวนขั้นตอนการทำงานขึ้นอยู่กับกำลัง 2 ของจำนวนข้อมูล อย่างในบางอัลกอริทึม ,จำนวนข้อมูลเพิ่มขึ้น 2 เท่า จำนวนงานเพิ่มขึ้น 4 เท่า บางอัลกอริทึมการเรียงลำดับเป็นชนิดนี้

## 5.2 การแก้ปัญหาโดยใช้การค้นหา

<u>เอเจนต์ที่แก้ไขปัญหา</u>

เอเจนต์ที่มีความฉลาดจะถูกสมมติเพื่อที่จะทำตัวในทางที่สิ่งแวดล้อมผ่านไปยังลำดับต่อเนื่องของสถานะที่มากที่สุดในการวัดประสิทธิภาพการทำงาน ในทางธรรมดา,ด้วยข้อจำกัดนี้ทำให้ยากที่จะแปลงให้มาอยู่ในการออกแบบเอเจนต์ได้อย่างสำเร็จ เช่นกับคำกล่าว, งานจะง่ายเพียงไรขึ้นอยู่กับว่าเอเจนต์สามารถแปลงเป้าหมาย (goal) และมุ่งไปที่จุดนั้น มาดูกันในตอนแรกว่าเอเจนต์ทำอย่างไรและทำไมเอเจนต์ถึงทำแบบนี้

ลองนึกถึงเอเจนต์ของเราในเมืองอะราด, โรมาเนีย, ไปยังจุดสิ้นสุดของทัวร์ในวันหยุด เอเจนต์มีตั๋วที่จะบินไปบูชาเรสในวันถัดมา ตั๋วไม่สามารถคืนเงินได้, วีซ่าของเอเจนต์ใกล้จะหมดอายุ และวันมะรืน, ไม่มีที่นั่งเหลือสำหรับ 6 สัปดาห์ ตอนนี้การวัดประสิทธิภาพของเอเจนต์ประกอบด้วยปัจจัยอื่น ๆ นอกเหนือจากราคาของตั๋วและความไม่พึงพอใจจากการถูกจับกุมและถูกขับไล่ออกมา ตัวอย่าง มันต้องการเพิ่มผิวสีแทน, เพิ่มความเป็นโรมาเนีย, อยู่ในสายตา,และอื่น ๆ ปัจจัยทั้งหมดนี้อาจแนะนำอาร์เรย์ขนาดใหญ่สำหรับการกระทำที่เป็นไปได้ ได้ให้ความลำบากของสถานการณ์, อย่างไรก็ตาม, มันควรแปลงเป้าหมายของการขับไปยังบูชาเรส การกระทำที่เป็นผลให้ไม่สำเร็จในการไปบูชาเรสในเวลาสามารถถูก ยกเลิกโดยปราศจากการพิจารณาต่อไป เป้าหมายเป็นเหมือนกับตัวช่วยจัดการกระทำโดยกำหนดจุดประสงค์ไว้ว่าเอเจนต์กำลังพยายามทำให้สำเร็จ สูตรโกล (goal formulation), อยู่บนพื้นฐานของสถานการณ์ปัจจุบันว่าเป็นขั้นตอนแรกในการแก้ไขปัญหา เช่นเดียวกับสร้างสูตรโกล, เอเจนต์อาจต้องการ ตัดสินใจในปัจจัยอื่น ๆ บางตัวว่ามีผลกระทบต่อความสามารถพอใจของทางที่ต่างกันที่จะไปยังเป้าหมาย

เราพิจารณาเป้าหมายเป็นกลุ่มของสถานะต่าง ๆ ในโลกจริง สถานะเหล่านั้นที่โกลพอใจ การกระทำสามารถถูกเห็นเป็นเหมือนกับสาเหตุการย้ายระหว่างสถานะ, ดังนั้นจึงเห็นได้ว่าเอเจนต์ต้องหาการกระทำซึ่งจะได้ทำให้มันไปยังสถานะที่เป็นโกล ก่อนที่มันทำเช่นนี้ได้, มันต้องการตัดสินใจการจัดลำดับการกระทำและสถานะที่นำมาพิจารณา ถ้ามันพยายามพิจารณาการกระทำที่ระดับชั้นของ "เคลื่อนย้ายเท้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซ้ายไปข้างหน้า 18 นิ้ว" หรือ "หมุนล้อไป 6 องศาทางซ้าย" มันอาจไม่ได้เจอทางออกจากการจอดจำนวนมาก, ปล่อยให้ไปบูชาเรสคนเดียว, เพราะว่าการสร้างวิธีแก้ปัญหาที่ระดับชั้นของรายละเอียดอาจจะเป็นปัญหาที่แก้ยาก การทำสูตรให้ปัญหา (problem formulation) เป็นตัวประมวลผลของการตัดสินใจว่าจะทำการกระทำใดและส่ถานะที่จะทำการพิจารณา และตามไปยังเป้าหมายจะพิจารณาถึงการกระทำที่ระดับชั้นของการขับรถจากเมืองใหญ่เมืองหนึ่งไปยังเมืองอื่น สถานะจะพิจารณาทางที่จะไปยังสถานะเฉพาะ

เอเจนต์ของเราได้แปลงเป้าหมายจากบูชาเรส, และมันกำลังพิจารณาเมืองที่จะขับจากอะราด มีถนน 3 สายที่ออกจากอะราด ทางหนึ่งไปยังสิบิว, ทางหนึ่งไปยังทิมิโสเอรา, และอีกทางหนึ่งไปยังซีรินด์ ไม่มีทางไหนไปยังโกล, ดังนั้นถ้าเอเจนต์ไม่คุ้นเคยกับเส้นทางในโรมาเนีย, มันอาจไม่รู้ว่าถนนที่จะไป ในอีกทางหนึ่ง, เอเจนต์จะไม่รู้ว่าทางไหนที่ทางที่เป็นไปได้และดีที่สุด, เพราะว่ามันไม่รู้พอว่าสถานะที่เป็นผลจากการทำแต่ละการกระทำ ถ้าเอเจนต์ไม่มีความรู้เพิ่มเติม, มันก็จะติดอยู่กับที่ ทางที่ดีสุดที่มันสามารถทำคือเลือกทางใดทางหนึ่งแบบสุ่ม

แต่สมมติว่าเอเจนต์มีแผนที่ของโรมาเนีย, ทั้งที่อยู่ในกระดาษหรือที่อยู่ในความจำ จุดของแผนที่ได้ให้ข้อมูลเอเจนต์เกี่ยวกับสถานะ มันอาจจะพาตัวเองไปยังจุดนั้น, และการกระทำที่มันใช้ เอเจนต์สามารถใช้ข้อมูลนี้เพื่อพิจารณาขั้นตอนในลำดับถัดไปของการเดินทางโดยสมมติผ่านไปยังแต่ละเมืองทั้ง 3 เมือง เพื่อพยายามหาเส้นทางที่ไปยังบูชาเรส ครั้งหนึ่งมันได้พบเส้นทางบนแผนที่จากอะราดไปยังบูชาเรส, มันสามารถไปถึงยังโกลของมันโดยเอาการกระทำในการขับรถว่าเกี่ยวข้องเป็นส่วนหนึงในเส้นทางในทางปกติ, เอเจนต์กับตัวเลือกมากมายของค่าที่ไม่ทราบสามารถตัดสินใจว่าควรทำอะไรโดยเริ่มทดลองลำดับต่อเนื่องของการกระทำที่เป็นไปได้แตกต่างกันที่นำไปยังสถานะที่รู้ค่า, และถัดมาเลือกทางที่ดีที่สุดมาหนึ่งทาง การประมวลผลของการดูลำดับที่ต่อเนื่องนี้ถูกเรียกว่า ค้นหา (search) อัลกอริทึมการค้นหานำปัญหาเป็นอินพุตและส่งค่าทางแก้ปัญหาในรูปแบบของลำดับต่อเนื่องของการกระทำ เมื่อพบทางแก้ปัญหาแล้ว, การกระทำที่เอเจนต์แนะนำสามารถนำไปใช้ได้ นี่เรียกว่า ส่วนการปฏิบัติ เรามีการออกแบบ "สร้างสูตร, ค้นหา, ปฏิบัติ" ที่ง่ายสำหรับเอเจนต์ ดังเช่นแสดงในรูปที่ 5-1 หลังจากสร้างสูตรโกลและปัญหาเพื่อหาทางแก้, เอเจนต์เรียกฟังก์ชันกาค้นหาเพื่อแก้ปัญหา มันใช้วิธีแก้ปัญหาเป็นตัวนำทางในการกระทำของมัน, ไม่ว่าจะทำอะไรที่วิธีแก้ปัญหาแนะนำเป็นเหมือนกับสิ่งถัดไปที่จะทำ, และเอาออกจากขั้นตอนจากลำดับต่อเนื่อง เมื่อวิธีแก้ปัญหาได้นำไปปฏิบัติแล้ว, เอเจนต์จะหาโกลอันใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function Simple-Problem-Solving-Agent (p) returns as action
    inputs: p, a percept
    static: s, an action sequence, initially empty
            state, some description of the current world state
            g, a goal, initially null
            problem, a problem formulation
    state      Update-State(state, p)
    if s is empty then
            g  <-- Formulate-Goal(state)
            problem <--Formulate-Problem(state,g)
            s  <-- Search(problem)
    action <-- Recommendation(s,state)
    s <--Remainer(s,state)
    return action
```

*รูปที่ 5-1 เอเจนต์แก้ไขปัญหาแบบง่าย*

## 5.3 การค้นหาทางที่ใช้สำหรับแก้ปัญหา

### 5.3.1สร้างลำดับของการกระทำ

เพื่อแก้ปัญหาการหาเส้นทางจากอะราดไปยังบูชาเรส, ตัวอย่างเช่น เราเริ่มจากสถานะเริ่มต้น, อะราด ขั้นตอนแรกคือการทดสอบว่าใช่สถานะโกลหรือไม่ แน่นอนอยู่แล้วว่ามันไม่ใช่, แต่มันสำคัญที่จะตรวจสอบเราสามารถแก้ปัญหาเล็กน้อยเช่น "เริ่มจากอะราด, ไปยังอะราด" เพราะว่านี่ไม่ใช่สถานะโกล, เราต้องการพิจารณาสถานะอื่นบางสถานะ นี่สามารถถูกทำได้โดยการให้ตัวปฏิบัติการไปยังสถานะที่อยู่ขณะนั้น จากนั้นก็สร้างกลุ่มของสถานะขึ้นมาใหม่ กระบวนการนี้เรียกว่า การขยาย (expanding) สถานะในกรณีนี้, เราสามารถไปยังสถานะใหม่ได้ 3 ทาง, "ในสิบิว","ในทิมิโสเอรา", และ "ในซีรานด์" เพราะว่ามีเส้นทางตรงไปขั้นตอนเดียวจากอะราดไปยังเมือง 3 เมือง ถ้ามีเพียงความเป็นไปได้เดียวที่เรายังไปต่อไปได้ แต่เมื่อไรก็ตามที่มีความเป็นไปได้หลายทาง, เราต้องทำทางเลือกเกี่ยวกับทางใดเพื่อพิจารณาต่อไป

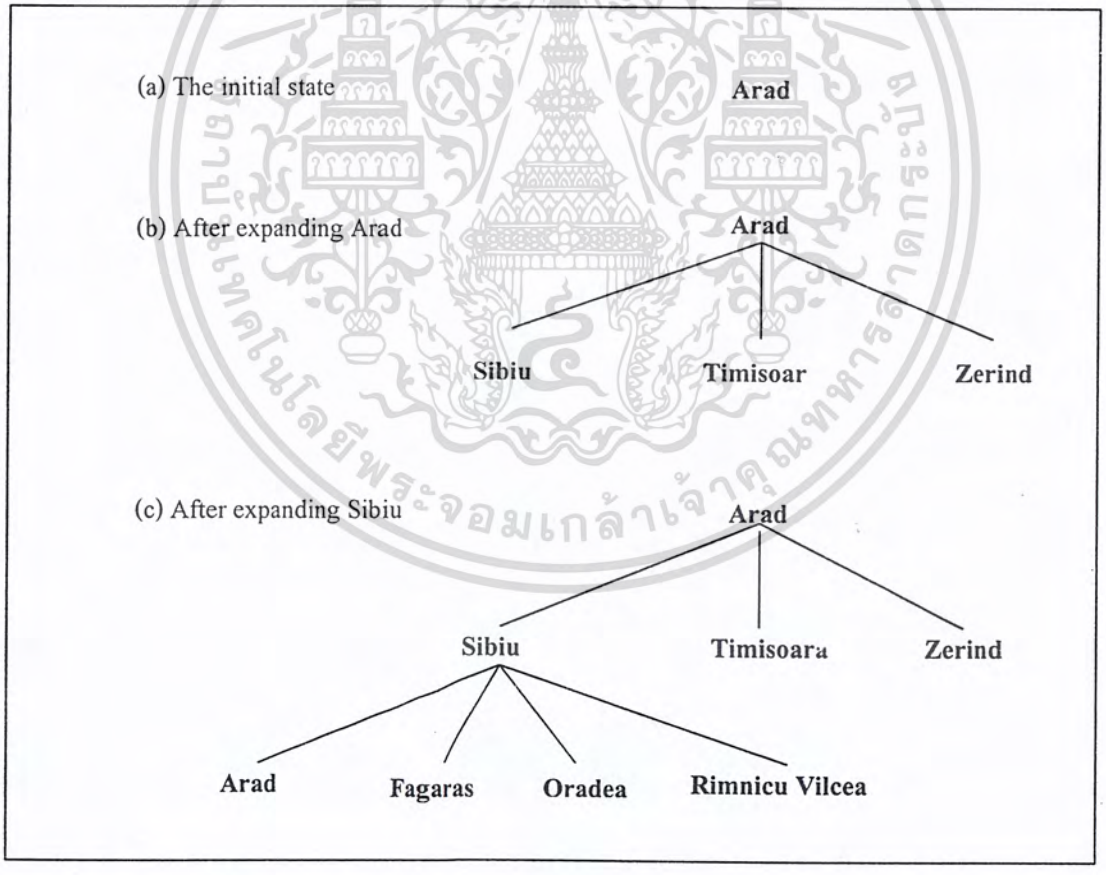
นี่เป็นการค้นหาเพียงอย่างเดียว----การเลือกทางเลือกหนึ่งและการวางอื่น ๆ อยู่ด้านข้างสำหรับทำทีหลัง, ในกรณีทางลือกอันแรกไม่นำไปสู่วิธีแก้ปัญหา สมมติว่าเราเลือกซีเรนด์ เราตรวจสอบดูว่าถ้ามันเป็นสถานะโกล(มันไม่ใช่), และถัดมาขยายมันให้รับ "ในอะราด" และ "ในโอราเดีย" เราสามารถเลือกอะไรก็ได้ในสองสถานะนี้, หรือไม่ก็กลับไปเลือกสิบิวหรือทิมิโสเอรา เราเลือก, ทดสอบ, และขยายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จนกระทั่งจะพบทางแก้ปัญหา, หรือไม่ก็ไม่มีสถานะเหลือให้ถูกขยายอีกแล้ว ทางเลือกของสถานะที่ขยายเป็นอันดับแรกถูกตัดสินโดยการวางแผนการค้นหา (search strategy)

มันมีประโยชน์ที่คิดถึงกระบวนการค้นหาที่สร้างขึ้นเป็นทรีการค้นหา (search tree) นั่นคือถูกเขียนทับสถานะที่ว่าง รากของทรีการค้นหาเป็นโหนดการค้นหา (search node) นำมาใช้ร่วมกับสถานะเริ่มต้น ลีฟโหนดของทรีมีลักษณะสถานะที่ไม่มีตัวถัดมาในทรี, ทั้งคู่เพราะว่าพวกมันยังไม่ได้ถูกขยาย, หรือเพราะว่าพวกมันถูกขยาย, แต่ถูกสร้างเป็นกลุ่มที่ว่างเปล่า ที่แต่ละขั้นตอน, อัลกอริทึมการค้นหาเลือกลีฟโหนดมาหนึ่งโหนดเพื่อมาขยาย รูปที่ 5-2 แสดงการขยายบางส่วนในทรีการค้นหาสำหรับค้นหาเส้นทางจากอะราดไปยังบูชาเรส อัลกอริทึมการค้นหาทั่วไป 'ถูกอธิบายอย่างไม่เป็นทางการในรูปที่ 5-3

มันสำคัญที่จะแบ่งระหว่างช่องว่างของสถานะและทรีการค้นหา สำหรับปัญหาการค้นหาเส้นทาง, มีเพียง 20 สถานะเท่านั้นที่อยู่ในที่ว่างของสถานะ, แต่ละเมืองใช้ 1 สถานะ แต่มีจำนวนทางเดินไม่จำกัดในที่ว่างของสถานะนี้, ดังนั้นทรีการค้นหามีจำนวนโหนดไม่จำกัด ตัวอย่างเช่น ในรูปที่ 5-2 , กิ่งของ อะราด-สิบิว-อะราด ต่อไปเป็น อะราด-สิบิว-อะราด-สิบิว-อะราด,และไปเรื่อย ๆ ไม่มีวันจบ สังเกตว่า อัลกอริทึมการค้นหาที่ดีหลีกเลี่ยงการวนอย่างข้างต้นนี้




*รูปที่ 5-2 ทรีการค้นหาบางส่วนสำหรับการหาเส้นทางจากอะราดไปยังบูชาเรส*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Function General-Search(problem, strategy) returns a solution, or failure
    Intitialize the search tree using the intitial state of problem
    Loop do
        If there are no candidates for expansion then return failure
        Choose a leaf node for expansion according to strategy
        If the node contains a goal state then return the corresponding solution
        Else expand the node and add the resulting nodes to the search tree
    end
```

*รูปที่ 5-3 คำอธิบายอย่างไม่เป็นทางการของอัลกอริทึมการค้นหาแบบธรรมดา*

### 5.3.2 โครงสร้างข้อมูลสำหรับทรีการค้นหา

มีหลายทางที่จะแสดงโหนด, แต่ในตอนนี้, เราจะกล่าวสมมติว่าโหนดเป็นโครงสร้างข้อมูลด้วยองค์ประกอบ 5 อย่าง :

- สถานะในที่ว่างของสถานะเพื่อโหนดที่ลักษณะเหมือนกัน
- โหนดในทรีการค้นหาที่สร้างโหนดนี้ (นี่เรียกว่าโหนดพ่อ-แม่)
- ตัวปฏิบัติการที่ถูกใช้ในการสร้างโหนด
- จำนวนโหนดบนทางเดินจากรากไปยังโหนดนี้ (ความลึกของโหนด)
- คอสของทางเดินจากสถานะเริ่มต้นไปยังโหนด

ชนิดข้อมูลของโหนดเป็นดังนี้ :

```
Datatype node
    Components: State, Parent-Node, Operator, Depth, Path-Cost
```

มันสำคัญที่จะจำไว้ว่าความแตกต่างระหว่างโหนดกับสถานะ โหนดเป็นตัวเก็บโครงสร้างข้อมูลที่ใช้แสดงทรีการค้นหาสำหรับตัวแทนปัญหาเฉพาะ ถูกสร้างโดยอัลกอริทึมเฉพาะอย่าง สถานะแสดงรูปร่าง (หรือกลุ่มของรูปร่าง)ของโลกจริง แม้โหนดมีความลึกและพ่อ-แม่ ที่สถานะไม่มี (นอกเหนือจากนี้, มันค่อนข้างจะเป็นไปได้สำหรับ 2 โหนดที่แตกต่างกันเก็บสถานะเดียวกัน, ถ้าสถานะนั้นถูกสร้างผ่านลำดับต่อเนื่องของการกระทำที่แตกต่างกัน 2 อย่าง) ฟังก์ชันขยายทำในส่วนสำหรับการคำนวณแต่ละองค์ประกอบของโหนดที่มันสร้างขึ้น

เรายังต้องการแสดงตัวเก็บโหนดที่กำลังรอที่จะถูกขยายจากกลุ่มนี้ แม้ว่านี้เป็นเพียงการเดินหน้าในทางทฤษฎีเท่านั้น, มันอาจเกินไปอย่างที่คำนวณ เพราะว่าฟังก์ชันวางแผนอาจต้องดูทุกองค์ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของกลุ่มที่เลือกทางที่ดีมาหนึ่งทาง แม้ว่า, เราจะสมมติว่าตัวเก็บโหนดว่าใช้เป็นคิว การทำงานบนคิวเป็นดังต่อไปนี้ :

- Make-Queue(*Elements*) สร้างคิวตามองค์ประกอบที่ให้มา
- Empty?(*Queue*) ส่งค่าจริงเมื่อไม่มีองค์ประกอบเหลือในคิว
- Remove-Front(*Queue*) นำองค์ประกอบออกมาที่ด้านหน้าของคิวและส่งค่านี้ออกไป
- Queuing-Fn(*Elements*,*Queue*) แทรกกลุ่มขององค์ประกอบเข้าไปในคิว หลากหลายวิธีของฟังก์ชันการคิว เพื่อสร้างหลากหลายวิธีของอัลกอริทึมการค้นหา

ด้วยนิยามที่ผ่านมา, เราสามารถเขียนเป็นแบบที่ถูกต้องของอัลกอริทึมการค้นหาแบบธรรมดา ที่แสดงในรูปที่ 5-4

```
Function General-Search(problem, Queuing-Fn) returns a solution, or failure

    Node ← Make-Queue(Make-Node(Initial-State[problem]))
    Loop do
        If nodes is empty then return failure
        Node ← Remove-Front(nodes)
        If Goal-Test[problem] applied to State(node) succeeds then return node
        Nodes ← Queuing-Fn(nodes,Expand(node,Operators[problem]))
    end
```

***รูปที่ 5-4 อัลกอริทึมการค้นหาแบบธรรมดา (สังเกตว่า Queue-Fn เป็นตัวแปรที่ค่าของมันจะเป็นฟังก์ชัน)***

## 5.4 การวางแผนการค้นหา

งานส่วนใหญ่ในส่วนของการค้นหาได้หาวิธีการวาแผนการค้นหาที่ถูกต้องสำหรับปัญหานั้น ๆ ในการศึกษาของเราจะศึกษาในส่วนของการประเมินแต่ละวิธีในเทอมทั้ง 4 ต่อไปนี้ :

- **Completeness** (ความสมบูรณ์) : เป็นการวางแผนที่ยืนยันว่าหาทางแก้ปัญหาได้เมื่อมีทางแก้ปัญหาเกิดขึ้นอยู่หนึ่งทางแล้ว
- **Time complexity** (ความซับซ้อนของเวลา) : ใช้เวลานานเท่าใดในการหาทางแก้ปัญหา?
- **Space complexity** (ความซับซ้อนของเนื้อที่หน่วยความจำ) : ใช้หน่วยความจำมากเท่าไรในการทำการค้นหา
- **Optimalitty** (ทางที่ดีที่สุด) : การวางแผนที่ใช้สามารถหาทางแก้ปัญหาที่มีคุณภาพสูงสุดหรือไม่เมื่อมีทางแก้ปัญหาหลาย ๆ ทาง?

การวางแผนการค้นหานั้นมีหลายวิธีที่แตกต่างกันไป แต่ในที่นี้จะนำมากล่าวเพียง 4 วิธี คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5.4.1 เบรดท์-เฟิรสท์ เซิรสช์ (Breadth-first search)**

วิธีการค้นหาที่ง่ายอันหนึ่งคือ เบรดท์-เฟิรสท์ เซิรสช์ ในวิธีนี้ โหนดรากถูกขยายเป็นอันดับแรก, ต่อมาทุกโหนดถูกสร้างออกมาจากโหนดรากในลำดับถัดมา, ถัดมาก็ตัวที่สืบทอดต่อมา ปกติแล้ว, ทุกโหนดที่ความลึก d ในทรีการค้นหาถูกขยายก่อนโหนดที่อยู่ที่ความลึก d+1 เบรดท์-เฟิรสท์ เซิรสช์ สามารถถูกปรับปรุงโดยเรียกอัลกอริทึมการค้นหาแบบทั่วไปกับฟังก์ชันคิวที่ใส่สถานะที่สร้างขึ้นมาใหม่ที่ปลายของคิว, หลังจากสถานะที่สร้างขึ้นมาก่อนหน้านี้

**Function** Breadth-First-Search(*problem*) **returns** a solution or failure

**Return** General-Search(*problem*, Enqueue-At-End)

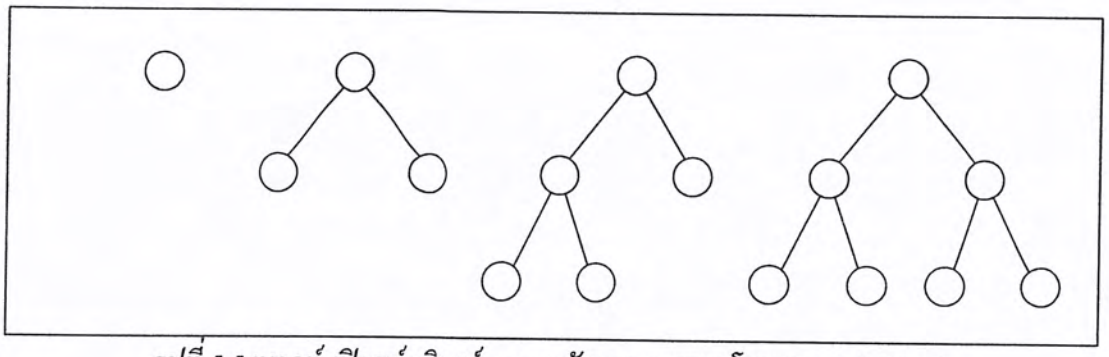
เบรดท์-เฟิรสท์ เซิรสช์ เป็นวิธีที่มีระบบเพราะว่ามันพิจารณาทางเดินความยาว 1 เป็นอันดับแรก, จากนั้นทุกโหนดที่ความยาว 2, และไปเรื่อย ๆ รูปที่ 5-5 แสดงการดำเนินการของการค้นหาบนทรีฐาน 2 (binary tree) อย่างง่าย ถ้ามีทางไปยังโกลได้, เบรดท์-เฟิรสท์ เซิรสช์ การันตีว่าหาได้พบแน่นอน, และถ้ามีหลาย ๆ ทางที่ไปยังโกลได้, เบรดท์-เฟิรสท์ เซิรสช์ จะหาทางที่ไปยังโกลสั้นที่สุดก่อน ในเทอมของตัววัดประสิทธิภาพที่สำคัญทั้ง 4, เบรดท์-เฟิรสท์ เซิรสช์ สามารถหาทางแก้ปัญหาได้อย่างสมบูรณ์ครบถ้วน, และมันช่วยหาทางที่ดีให้กับค่าของทางเดินที่ไม่เป็นฟังก์ชันลดลงของความลึกของโหนด (เงื่อนไขนี้ถูกทำให้จริงก็ต่อเมื่อตัวปฏิบัติการทุกตัวมีค่าเท่ากัน)

เรื่องต่อมา, ข่าวดีเกี่ยวกับ เบรดท์-เฟิรสท์ เซิรสช์ ดูว่าทำไมมันไม่เป็นวิธีที่ควรเลือก, เราต้องพิจารณาจำนวนเวลาและหน่วยความจำที่มันใช้ในการทำการค้นหาให้เสร็จสมบูรณ์ เพื่อที่จะดูสิ่งต่อไปนี้, เราพิจารณาที่ว่างของสถานะที่สมมติขึ้นมาที่ทุก ๆ สถานะสามารถถูกขยายออกเป็น b สถานะใหม่ เราพูดว่าปัจจัยการแตกโหนดลูก (branching factor) ของสถานะพวกนี้ (และของทรีการค้นหา) คือ b รากของทรีการค้นหาสร้าง b โหนดที่ระดับชั้นที่ 1, แต่ละตัวก็สร้าง b โหนดขึ้นมาอีก, สำหรับจำนวนทั้งหมดคือ $b^2$ ที่ระดับชั้นที่ 2 แต่ละโหนดที่ชั้นที่ 2 ก็สร้าง b โหนดอีก, เพิ่มขึ้นมาอีก $b^3$ โหนดที่ระดับชั้นที่ 3 ,และเป็นเช่นนี้ไปเรื่อย ๆ ตอนนี้สมมติว่าทางแก้ของปัญหานี้มีความยาวทางเดินเท่ากับ d ดังนั้นจำนวนโหนดที่มากที่สุดที่จะขยายก่อนจะพบโหนดโกล คือ

$$1 + b + b^2 + b^3 + \ldots + b^d$$

นี่เป็นจำนวนที่มากที่สุด, แต่โกลสามารถถูกพบได้ทุกจุดในระดับชั้นที่ d ในกรณีที่ดีที่สุด, ควรจะมีจำนวนน้อยกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



*รูปที่ 5-5 เบรดท์-เฟิรสท์ เซิรสช์ trees หลังจากการขยายโหนด 0, 1, 2, และ 3*

สำหรับคนที่ต้องวิเคราะห์ความซับซ้อนต้องปวดหัว (หรือ ตื่นเต้น, ถ้าพวกเขาเป็นคนประเภทชอบความท้าทาย) เมื่อไหร่ก็ตามที่พวกเขาเห็นความซับซ้อนแบบเอ็กโปเนนเชี่ยล เช่น $O(b^d)$

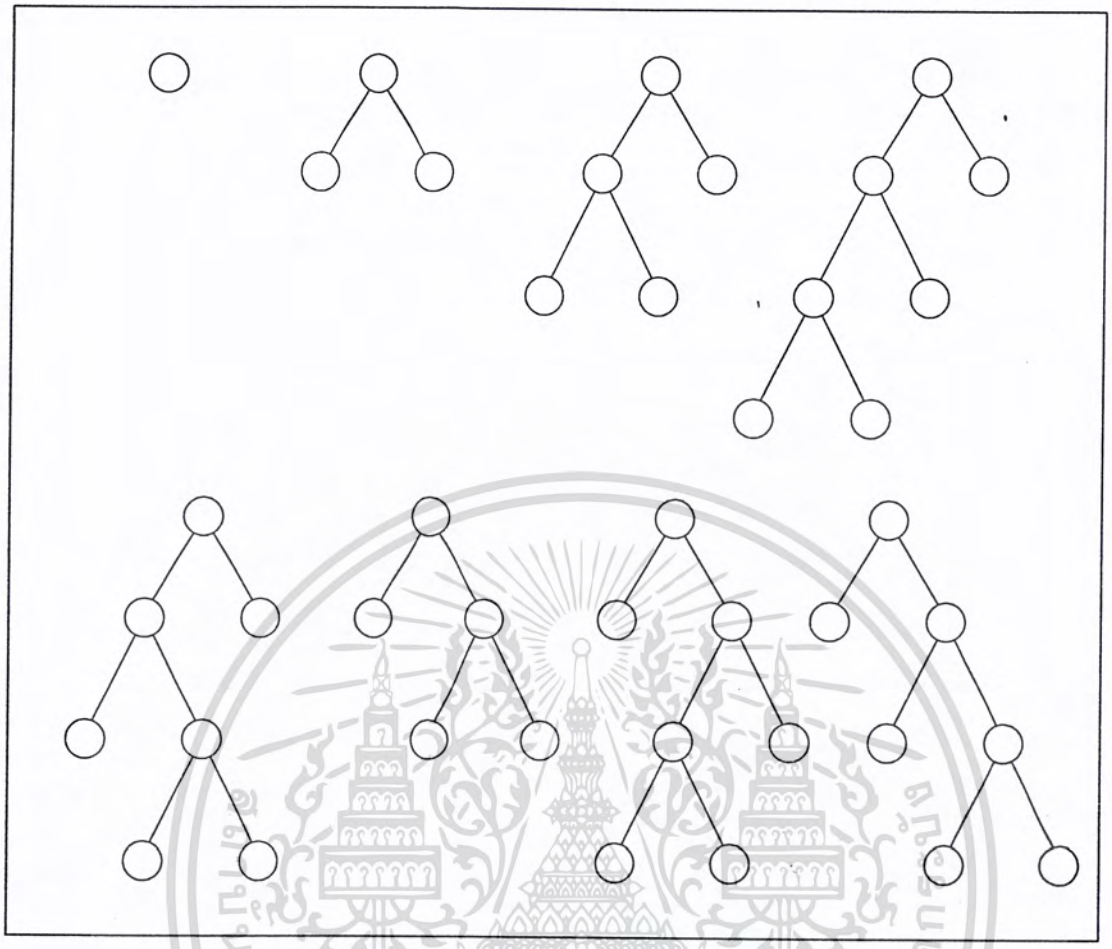
### 5.4.2 เดพท์–เฟิรสท์ เซิรสช์ (Depth–first search)

เดพท์-เฟิรสท์ เซิรสช์ จะขยาย 1 โหนดใด ๆ ไประดับชั้นที่ลึกที่สุดของทรี เมื่อการค้นหาเจอกับทางตัน (โหนดที่ไม่ใช่โกลไม่มีการขยาย) ทำการค้นหากลับไปและขยายโหนดที่ระดับชั้นที่สั้นกว่า วิธีนี้ถูกใช้โดยการค้นหาแบบธรรมดาด้วยฟังก์ชันคิวที่ใส่สถานะที่สร้างขึ้นมาใหม่ที่ด้านหน้าของคิว เพราะว่าโหนดที่ถูกขยายอยู่ลึกที่สุด, โหนดลูกของมันจะอยู่ลึกกว่าและไปจนถึงชั้นที่ลึกสุด กระบวนการของการค้นหาอยู่ในรูปที่ 5-6

เดพท์-เฟิรสท์ เซิรสช์ มีความต้องการหน่วยความจำน้อยมาก เหมือนกับรูปที่แสดงมันต้องการที่เก็บเพียงแค่ทางเดินเดียวจากรากที่ลีฟโหนด, อย่างเดียวโดยยังคงไม่ขยายโหนดพี่น้องสำหรับแต่ละโหนดบนทางเดิน สำหรับที่ว่างของสถานะกับปัจจัยการแตกโหนดลูก b และความลึกที่มากที่สุด m, เดพท์-เฟิรสท์ เซิรสช์ ต้องการที่เก็บเพียง bm โหนด, ในทางตรงกันข้ามกับ $b^d$ ที่ต้องใช้ใน เบรดท์-เฟิรสท์ เซิรสช์

ความซับซ้อนของเวลาสำหรับ เดพท์-เฟิรสท์ เซิรสช์ คือ $O(b^m)$ สำหรับปัญหาที่มีหลาย ๆ ทางแก้, เดพท์-เฟิรสท์ อาจจะเร็วกว่า เบรดท์-เฟิรสท์, เพราะว่ามันมีโอกาสที่จะพบโกลหลังจากสำรวจเพียงส่วนหนึ่งของเนื้อที่ทั้งหมด เบรดท์-เฟิรสท์ เซิรสช์ ยังคงต้องดูที่ทุกทางเดินที่ความยาว d – 1 ก่อนที่จะพิจารณาที่ความยาวใด ๆ d เดพท์-เฟิรสท์ เซิรสช์ ยังคงเป็น $O(b^m)$ ในกรณีที่แย่ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



*รูปที่ 5-6 เดพท์-เฟิรสท์ เซิรสช์ทรีสำหรับทรีฐาน 2 โหนดที่ความลึก 3 ถูกสมมติว่าไม่มีโหนดลูก*

การย้อนกลับของ เดพท์-เฟิรสท์ เซิรสช์ มันสามารถติดในการลงไปในทางที่ผิด หลาย ๆ ปัญหามีทรีการค้นหาที่มีความลึกมากหรือไม่มีจุดสิ้นสุดเลยก็ได้, ดังนั้น เดพท์-เฟิรสท์ เซิรสช์ จะไม่สามารถแก้ไขจากการเลือกที่โชคไม่ดีที่โหนดใดโหนดหนึ่งใกล้ ๆ กับระดับบนสุดของทรี การค้นหาจะลงต่อไปโดยไม่มีการกลับขึ้นมา, แม้ว่าโกลอยู่ไม่ลึก, แม้ว่าปัญหาเหล่านี้ เดพท์-เฟิรสท์ เซิรสช์ ทั้งยังคงติดอยู่ในลูปที่ไม่มีจุดสิ้นสุดและไม่ส่งค่าผลลัพธ์ทางแก้ปัญหากลับมา, หรือมันอาจพบทั้งทางไปยังโกลที่ยาวกว่าทางที่ดีที่สุด นั่นหมายถึง เดพท์-เฟิรสท์ เซิรสช์ ทั้งไม่สมบูรณ์และไม่ให้ค่าที่ดีที่สุด เพราะว่าสิ่งนี้, เดพท์-เฟิรสท์ เซิรสช์ ควรหลีกเลี่ยงสำหรับทรีการค้นหาที่ใหญ่หรือมีความลึกไม่สิ้นสุด

มันเป็นข้อเล็กน้อยที่ใช้ เดพท์-เฟิรสท์ เซิรสช์ ด้วยการค้นหาแบบทั่วไป

**Function** Depth-First-Search(*problem*) **returns** a solution, or failure

General-Search(*problem*, Enqueue-At-Front)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มันเป็นพื้นฐานในการใช้ เดพท์-เฟิรสท์ เซิรสช์ ด้วยฟังก์ชันรีเคอซีฟ (recursive) ที่เรียกตัวเองบนแต่ละตัวลูกของมัน ในกรณีนี้, คิวถูกเก็บโดยนัยในสถานะพื้นฐานของแต่ละการเรียกร้องของสแตก

### 5.4.3 เดพท์-ลิมิต เซิรสช์ (Depth-limited search)

เดพท์-ลิมิต เซิรสช์ หลีกเลี่ยงการติดอยู่ในลูปของ เดพท์-เฟิรสท์ เซิรสช์ โดยได้กำหนดคัทออฟ (cut off) พบความลึกของทางเดินที่ลึกที่สุด คัทออฟตัวนี้สามารถถูกแก้ไขด้วยอัลกอริทึมพิเศษ เดพท์-ลิมิต เซิรสช์, หรือการใช้อัลกอริทึมการค้นหาแบบทั่วไปด้วยตัวปฏิบัติการที่เก็บเส้นทางของความลึก ตัวอย่าง, ในแผนที่ของโรมาเนีย, มี 20 เมือง, ดังนั้นเรารู้ว่าถ้ามีทางไปยังโกลที่มันต้องใช้ความยาว 19 ที่ระยะยาวที่สุด เราสามารถกำหนดคัทออฟของความลึกโดยการใช้ตัวปฏิบัติการของฟอร์ม "ถ้าเราอยู่ในเมือง A และมีเดินทางผ่านทางอย่างน้อย 19 ขั้นตอน, ดังนั้นสร้างสถานะใหม่ในเมือง B ด้วยความยาวทางเดินที่ยาวกว่าอันเดิน" ด้วยตัวปฏิบัติการอันใหม่เริ่มใช้, เรารับประกันที่พบโกลถ้ามันมี, แต่เรายังคงไม่รับประกันว่าพบทางที่สั้นที่สุดก่อนอันดับแรก: เดพท์-ลิมิต เซิรสช์ หาคำตอบได้ครบแต่ไม่ใช่ทางที่ดีที่สุด ถ้าเราเลือกจำกัดความลึกที่น้อยเกินไป, ดังนั้น เดพท์-เฟิรสท์ เซิรสช์ ไม่สามารถหาคำตอบได้ครบ ความซับซ้อนของเวลาและเนื้อที่ของ เดพท์-ลิมิต เซิรสช์ เหมือนกับของ เดพท์-เฟิรสท์ เซิรสช์ มันใช้เวลา $O(b^l)$ และเนื้อที่ O(bl), เมื่อ l คือตัวจำกัดความลึก

### 5.4.4 อิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ (Iterative deepening search)

ส่วนที่ยากเกี่ยวกับ เดพท์-ลิมิต เซิรสช์ คือ การเลือกตัวจำกัดที่ดี เราเลือก 19 เป็นเหมือน "สังเกต" ตัวจำกัดความลึกสำหรับปัญหาโรมาเนีย, แต่ในความเป็นจริงถ้าเราศึกษาแผนที่อย่างละเอียดแล้ว, เราพบว่าที่เมืองใด ๆ สามารถไปถึงจากเมืองอื่นใด ๆ ในส่วนใหญ่ 9 ขั้นตอน ตัวเลขนี้, รู้จักกันในชื่อของเส้นผ่าศูนย์กลางของที่ว่างของสถานะ อย่างไรก็ตาม, สำหรับปัญหาส่วนใหญ่, เราจะไม่รู้จักค่าที่ใช้กำหนดความลึกที่เหมาะสมจนกว่าเราจะแก้ปัญหานั้นได้แล้ว

อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ เป็นวิธีที่เลี่ยงปัญหาในการเลือกตัวกำหนดความลึกที่ดีที่สุดโดยการพยายามตัวกำหนดความลึกทุกตัวที่เป็นไปได้ : อันดับแรก ความลึก 0, ต่อมา ความลึก 1, ต่อมา ความลึก 2, และไปเรื่อย ๆ อัลกอริทึมถูกแสดงในรูปที่ ผลกระทบที่ได้รับใน, อิเทอเรทีฟ ดิพเพ็นนิ่ง คือ ได้รวมข้อดีของ เดพท์-เฟิรสท์ และ เบรดท์-เฟิรสท์ เซิรสช์ มันหาทางที่ดีที่สุดและครบสมบูรณ์, เหมือน เบรดท์-เฟิรสท์ เซิรสช์, แต่ใช้ความต้องการของหน่วยความจำน้อยเหมือน เดพท์-เฟิรสท์ เซิรสช์ ลำดับของการขยายสถานะคล้ายกับ เบรดท์-เฟิรสท์ เซิรสช์, ยกเว้นบางสถานะที่ถูกขยายหลายครั้ง รูปที่ 5-8 แสดงการทำซ้ำ 4 ครั้งของ อิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ ในทรีฐาน 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Function** Iterative-Deepening-Search(*problem*) **returns** a solution sequence

**Inputs**: *problem*, a problem

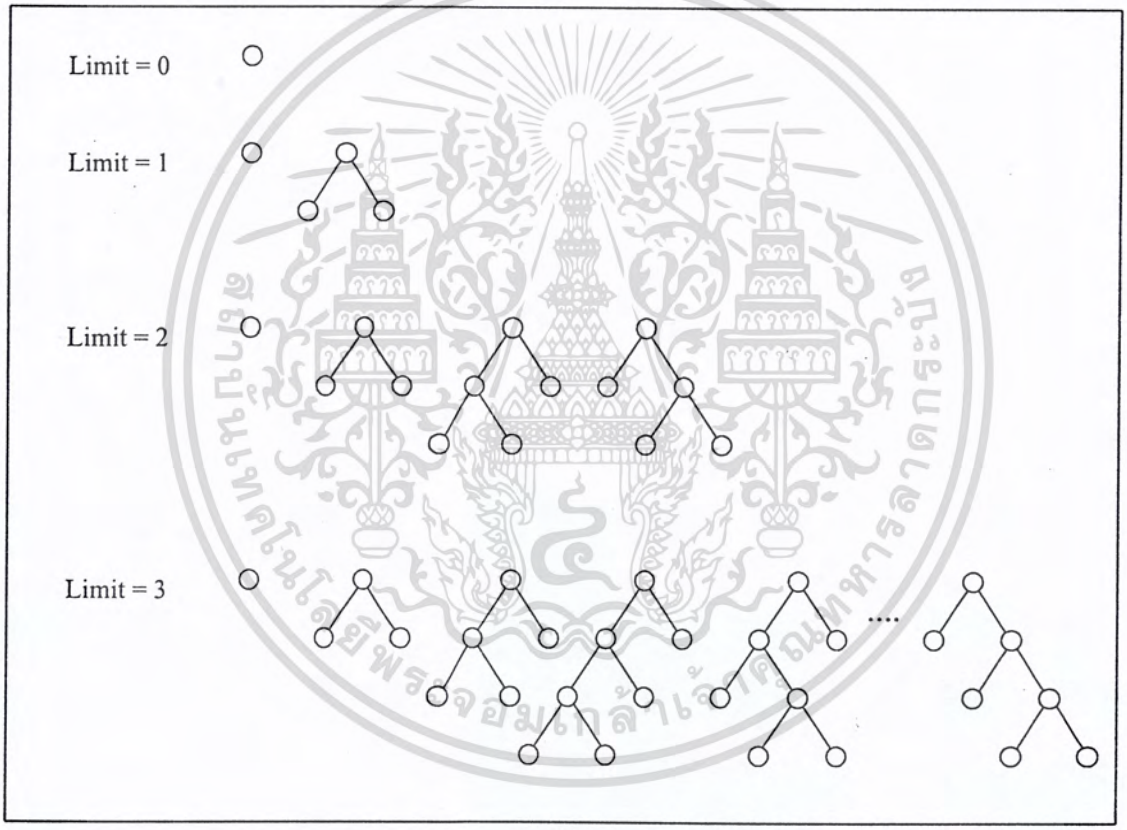
**For** *dept h*⟵ 0 **to** ∞ **do**

**If** Depth-Limited-Search(*problem*, *depth*) succeeds **then return** its result

**End**

**Return** failure

*รูปที่ 5-7 อัลกอริทึม อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์*




*รูปที่ 5-8 การเดินซ้ำ 4 ครั้งของ อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์*

อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ อาจดูเหมือนเป็นการทำที่สูญเปล่า, เพราะว่ามีหลายสถานะทีถูกขยายหลายครั้ง อย่างไรก็ตาม. สำหรับปัญหาส่วนใหญ่ ค่าการใช้งานของการทำการขยายหลายครั้งนั้นจริง ๆ แล้วค่อนข้างน้อย สิ่งที่เกิดขึ้นเอง, เหตุผลคือว่าในทรีการค้นหาแบบเอ็กโปเนนเชี่ยล, โหนดทั้งหมดส่วนใหญ่อยู่ที่ระดับชั้นด้านล่าง, ดังนั้นมันไม่เกิดผลมากในระดับชั้นที่สูงขึ้นไปจะมีการขยายหลายครั้ง การเรียกการขยายซ้ำใน เดพท์-ลิมิต เซิรสช์ ไปยังความลึก d โดยปัจจัยการแตกโหนดลูก b คือ

$$1 + b + b^2 + \ldots + b^{d-2} + b^{d-1} + b^d$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลองแทนค่าดู, ให้ b = 10 และ d = 5 ,ผลรวมคือ

$$1 + 10 + 100 + 1{,}000 + 10{,}000 + 100{,}000 = 111{,}111$$

ใน อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์, โหนดที่ระดับชั้นล่างถูกขยายเพียงครั้งเดียว, ในระดับถัดจากระดับล่างจะถูกขยาย 2 ครั้ง, และเป็นเช่นนี้ไปเรื่อย ๆ จนถึงที่รากของทรีการค้นหา, ที่ถูกขยายเป็นจำนวน d-1 ครั้ง ดังนั้นจำนวนรวมของการขยายใน อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ คือ

$$(d + 1)1 + (d)b + (d - 1)b^2 + \ldots + 3d^{b-2} + 2b^{d-1} + 1b^d$$

แทนค่าอีกครั้ง, ให้ b = 10 และ d = 5 ผลรวมคือ

$$6 + 50 + 400 + 3{,}000 + 20{,}000 + 100{,}000 = 123{,}456$$

รวมทั้งหมดเข้าด้วยกัน, อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ จาก ความลึก 1 ไปทุกทางลงไปยังความลึก d ขยายโหนดเพิ่มขึ้นเพียงประมาณ 11% เมื่อเปรียบเทียบกับ เบรดท์-เฟิรสท์ เซิรสช์ หรือ เดพท์-ลิมิต เซิรสช์ ที่ความลึก d, เมื่อ b = 10 ถ้าปัจจัยการแตกโหนดลูกเพิ่มขึ้น ค่าการใช้งานของการขยายสถานะแบบซ้ำลดลง แต่เมื่อปัจจัยการแตกโหนดลูกเท่ากับ 2, อิเทอเรทีฟ ดีพเพ็นนิ่ง เซิรสช์ ใช้ระยะเวลาเพียงประมาณ 2 เท่า เมื่อเทียบกับ เบรดท์-เฟิรสท์ เซิรสช์ ที่ครบสมบูรณ์ นั่นหมายถึงว่าความซับซ้อนของเวลาของ อิเทอเรทีฟ ดิพเพ็นนิ่ง ยังคงเป็น $O(b^d)$, และความซับซ้อนของที่ว่าง คือ O(bd) โดยทั่วไปแล้ว อิเทอเรทีฟ ดิพเพ็นนิ่ง ถูกนำไปใช้เป็นวิธีการค้นหาเมื่อมีที่ว่างของการค้นหา (search space) และ ความลึกของโกลไม่ทราบค่า

## 5.5 การเปรียบเทียบวิธีการค้นหา

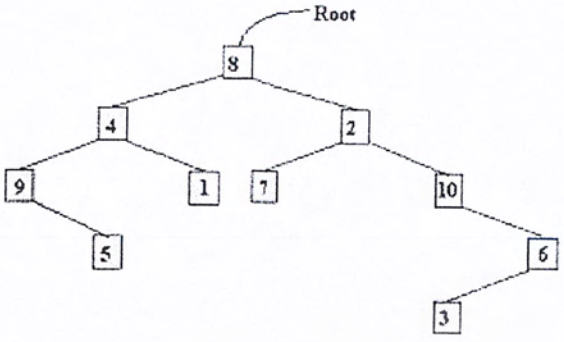
ตารางที่ 5-1 เป็นการเปรียบเทียบการค้นหาทั้ง 4 แบบในเทอมของเกณฑ์การประเมินทั้ง 4 ตัว

| Criterion | Breadth-First | Depth-First | Depth-Limited | Iterative Deepening |
|---|---|---|---|---|
| Time | $b^d$ | $b^m$ | $b^l$ | $b^d$ |
| Space | $b^d$ | bm | bl | bd |
| Optimal? | Yes | No | No | Yes |
| Complete? | Yes | No | Yes, if $l \geq d$ | Yes |

***ตารางที่ 5-1 การประเมินวิธีการค้นหา b เป็นปัจจัยการแตกโหนดลูก (branching factor); d เป็นความลึกของการแก้ปัญหา; m เป็นความลึกสูงสุดของทรีการค้นหา; l เป็นค่าความลึกที่กำหนดไว้***

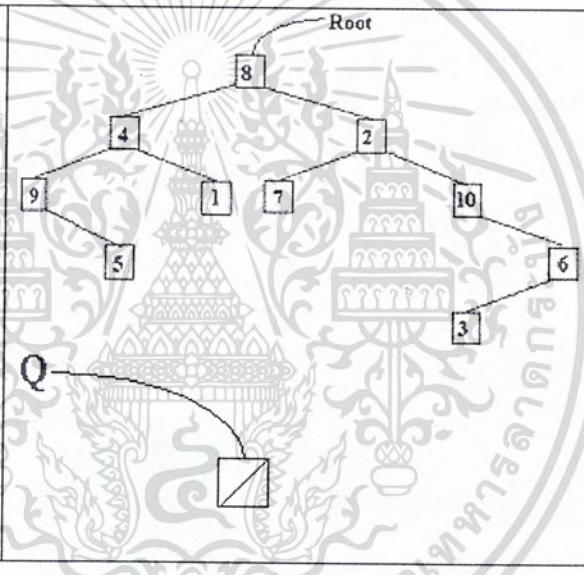
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5.6 ตัวอย่างการค้นหา เบรดท์-เฟิรสท์ เซิรสช์**




ขั้นตอนที่ 1

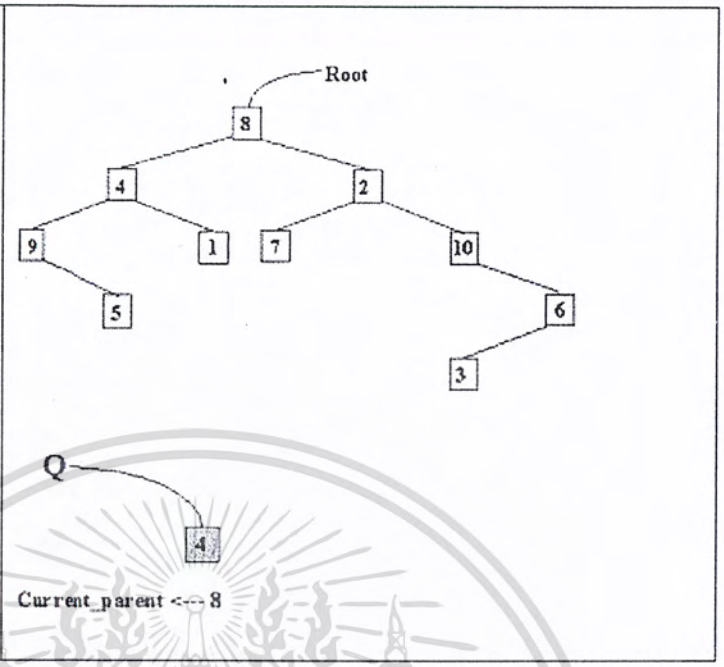
- ดูทุกโหนดตลอดทั้งทรีและกำหนดค่า visit ให้แต่ละโหนดเท่ากับ false
- แสดง visit = false โดยการใช้สีอ่อน
- เริ่มคิวเท่ากับคิวว่างเปล่า




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

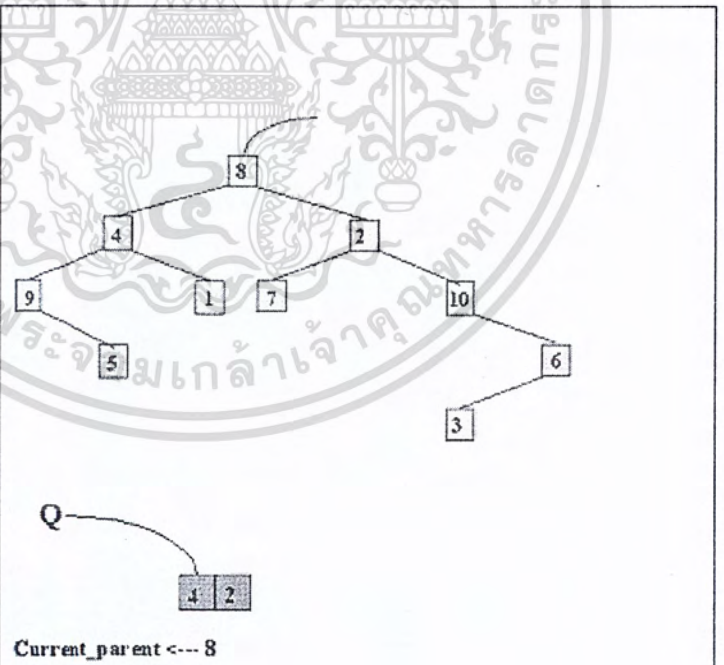
ขั้นตอนที่ 2

- หาโหนดที่ 7 จากในทรี ทรีว่างเปล่าหรือไม่?
- เริ่มการค้นหาที่โหนดราก โหนดมีค่าเท่ากับ 7 หรือไม่? ไม่ใช่ มันเท่ากับ 8
- กำหนดค่า visited [root] = true, แสดงให้เห็นโดยสี
- พุชโหนดราก (หมายเลข 8) ลงไปในคิว



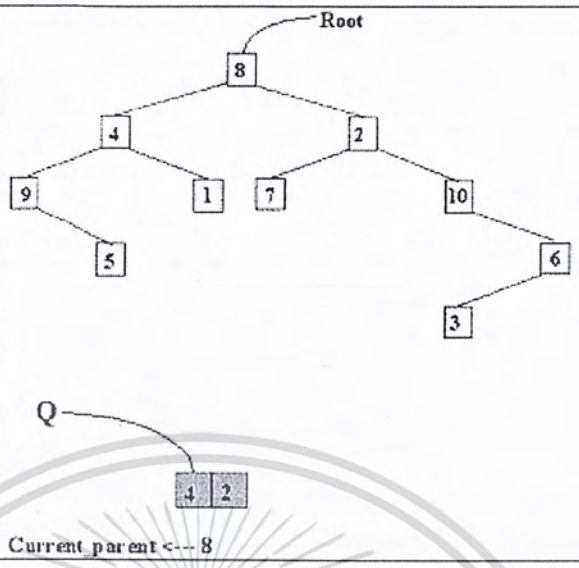

ขั้นตอนที่ 3

- เมื่อโหนดรากไม่มีโหนดพี่น้อง ก็ให้ไปยังโหนดลูก
- กำหนดค่า current_parent = top of Q (= 8)
- ป๊อป 8 ออกจากคิว
- ดูที่ current_parent.child child.visited = true หรือไม่? ไม่ใช่
- ให้ค่า child.visited = true (ดูที่สี)
- โหนดนี้ = 7 หรือไม่? ไม่ใช่มัน = 4
- พุชโหนดหมายเลข 4 ลงในคิว




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 4

| | |
|---|---|
| - ดูที่ลูกตัวต่อไปของ current_parent<br>- child.visited = true หรือไม่? ไม่ใช่<br>- โหนดนั้น = 7 หรือไม่? ไม่ใช่ มัน = 2<br>- ให้ค่า visited = true<br>- พุชโหนดหมายเลข 2 ลงไปในคิว |  |

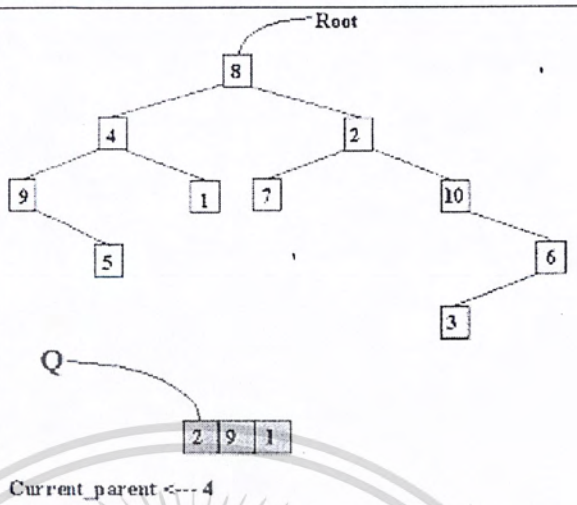
ขั้นตอนที่ 5

| | |
|---|---|
| - ดูที่โหนดตัวต่อไปของ current_parent<br>- child.visit = true หรือไม่? ใช่<br>- Q.empty = true หรือไม่? ไม่ใช่<br>- กำหนด current_parent top of queue (= 4)<br>- ป๊อป top ของ Q ออก<br>- ดูที่ current_parent.child<br>- โหนดนั้น = 7 หรือไม่? ไม่ใช่มัน = 9<br>- ตั้งค่า visited = true<br>- พุชโหนดหมายเลข 9 ลงไปบน Q | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

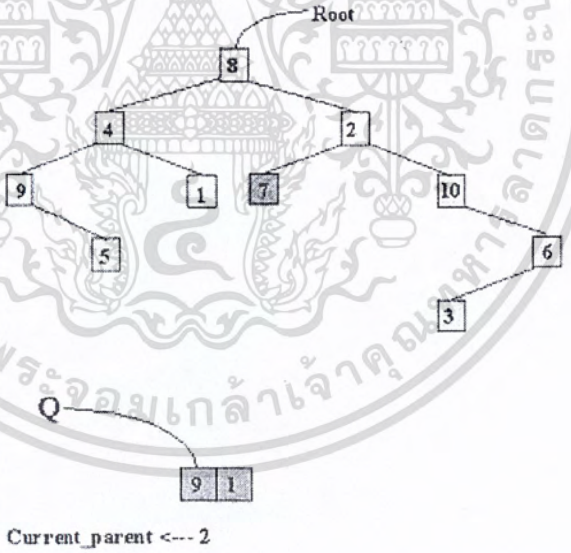
ขั้นตอนที่ 6

- ดูที่ลูกตัวต่อไปของ current_parent
- child.visited = true หรือไม่?
- มัน = 7 หรือไม่? ไม่ใช่มัน = 1
- ให้ค่า visited = true
- พุชโหนดหมายเลข 1 ลงไปในคิว




ขั้นตอนที่ 7

- ดูที่ลูกตัวต่อไปของ current_parent
- child.visited = true หรือไม่?
- Q.empty = true หรือไม่? ไม่ใช่
- กำหนดค่า current_parent = top ของ Q (= 2)
- ป๊อป top ของ Q (2) ออก
- โหนดนั้น = 7 หรือไม่? ใช่
- ส่งค่ากลับว่าพบแล้ว




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6
# ส่วนการติดต่อกับผู้ใช้

ในส่วนของการติดต่อกับผู้ใช้นั้น จะมีการเพิ่มข้อมูลในส่วนที่เป็นการติดต่อกับผู้ใช้ให้มีความสะดวกมากยิ่งขึ้น โดยมีการเพิ่มส่วนการพรีวิวแผนที่ก่อนจะทำการพิมพ์เข้ามาด้วย ส่วนการติดต่อกับผู้ใช้นี้เป็นส่วนที่สำคัญส่วนหนึ่งของโปรแกรม โดยจะใช้ภาษา จาวาในการเขียนเช่นเดียวกับ ส่วนแรก แต่จะต่างกันตรงที่ในส่วนนี้จะใช้ จาวาแอพเพล็ต ซึ่งจะมีคุณสมบัติในการติดต่อและแสดงผลกับผู้ใช้ได้ดีกว่าภาษา จาวาปกติที่ใช้ในการค้นหาข้อมูลในส่วนแรก

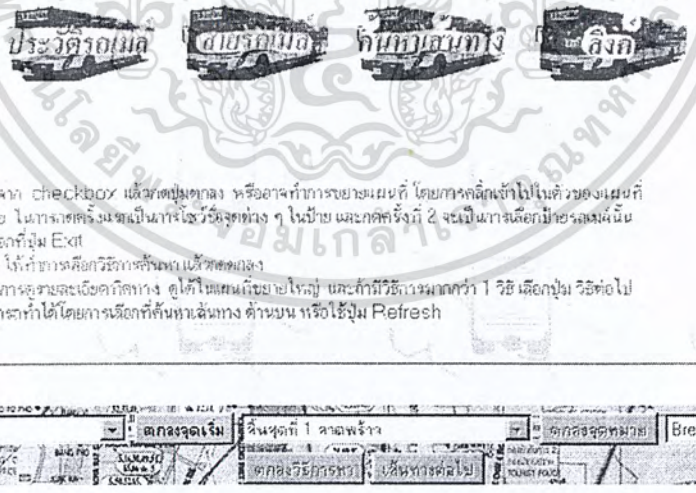
ในส่วนของการติดต่อกับผู้ใช้และการแสดงผลนี้ จะแบ่งย่อยได้เป็น 3 ส่วนย่อยคือ

1.ส่วนการรับอินพุตข้อมูล

2. ส่วนการแสดงผลข้อมูล

## 6.1 ส่วนการรับอินพุตข้อมูล

ในที่นี้เราจะกล่าวถึงเฉพาะส่วนของโปรแกรมที่เกี่ยวข้องกับการค้นหาข้อมูลเท่านั้น คือสนใจเฉพาะ หน้าเพจของเว็บเพจที่เกี่ยวข้องกับการค้นหาและแนะนำข้อมูลสายรถเมล์ จะมีการเปลี่ยนหน้าจอการอินพุต โดยดูตัวอย่างจากรูปจะพบว่ามีการส่วนของทางเลือกที่จะให้ใช้วิธีการใดในการเซิรสช์ และมีการอินพุตข้อมูลโหนดที่ใช้ค้นหา และส่วนของการเลือกพรีวิวการพิมพ์เข้ามาด้วย




*รูปที่ 6-1 แสดงหน้าจอการอินพุตข้อมูลในส่วนบน*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**6.1.1 การอินพุตข้อมูลส่วนบน**

โดยเราจะแบ่งการอินพุตข้อมูลเข้าไปในการค้นหาออกเป็น 4 ส่วนย่อย ส่วนที่ 1 คือรูปรถเมล์แต่ละรูป ส่วนที่ 2 คือ กล่องข้อความและ ปุ่ม ที่อยู่บน ตัวแผนที่ ส่วนที่ 3 คือส่วนของวิธีการใช้งาน และส่วนสุดท้าย คือส่วนที่ 4 นั้นคือส่วน ของตัวรูปแผนที่ที่อยู่ทางด้านล่าง

โดยแต่ละส่วนจะมีรายละเอียดต่าง ๆ ดังนี้

1. ส่วน ที่ 1 คือส่วนของการเลือกหน้าจอเพจต่าง ๆ ในเว็บเพจนี้โดยในส่วนนี้เราจะไม่ให้ความสนใจ เนื่องจากจะเป็นส่วนรายละเอียดต่าง ๆ ที่ไม่เกี่ยวข้องกับโปรแกรมขลุงเรา อาทิเช่น ประวัติรถเมล์ โดยเราสนใจในเพจย่อย คือตัว เพจย่อย ค้นหาเส้นทาง มากกว่า ซึ่งจะมีรายละเอียดเป็น ส่วนที่ 2 และส่วนที่ 3 ที่จะกล่าวถึงต่อไป

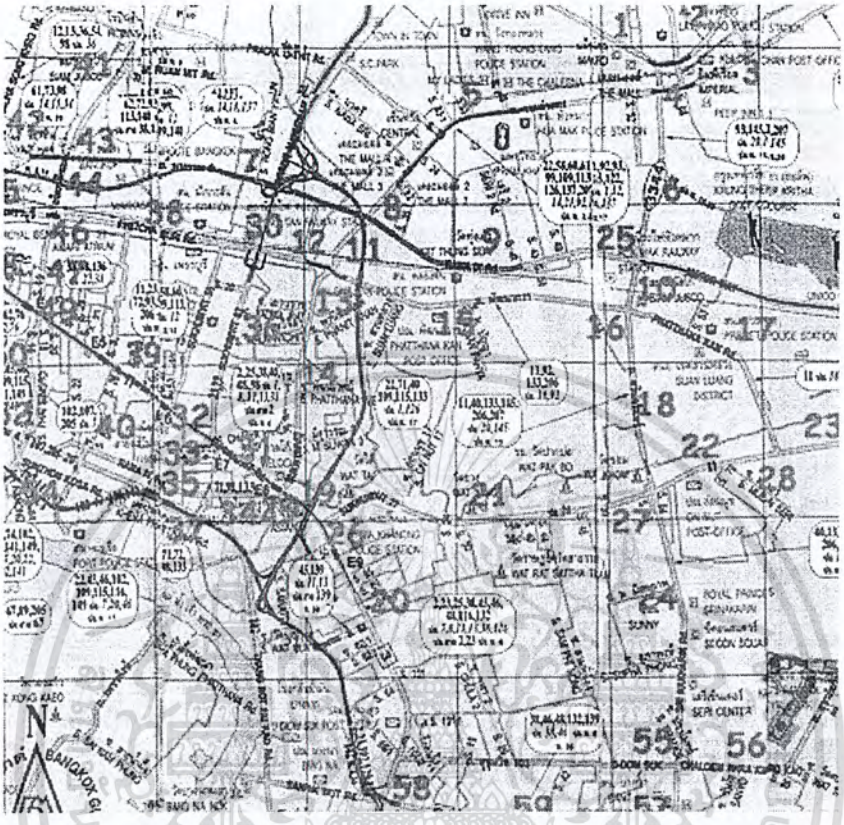
2. ส่วนที่ 2 คือส่วนที่ใช้ในการป้อนข้อมูล โดยเราจะทำเป็น เมนูรายชื่อของสถานที่ที่เราจะทำการค้นหา โดยเรากำหนดให้สถานที่แต่ละจุดเป็นป้ายรถเมล์หรือเป็นโหนด โดยในป้ายรถเมล์แต่ละป้ายจะแทนช่วงถนนและแยกในแต่ละแยก ซึ่งจะทำการเลือกได้จากเมนูของจุดที่จะขึ้นรถ แล้วหลังจากนั้นก็จะทำการเลือกจุดหมายปลายทางที่ต้องการจะลงจากเมนู อีกอันหนึ่ง ซึ่งเมื่อเลือกได้แล้วให้กด ตกลง เพื่อทำการยืนยันข้อมูล โดยในแต่ละเมนูลิสต์รายชื่อนี้ จะมีตัวเลขกำกับไว้ในแต่ละสถานที่ เช่นจากรูปกำหนดให้ ป้ายรถเมล์บนถนนลาดพร้าว มีเลขกำกับคือ เลข 1 หมายถึงป้ายรถเมล์หมายเลข 1 ถ้าเป็นป้ายรถเมล์บนถนนสุขาภิบาล 2 ก็จะเป็น เลข 2 ซึ่งมีบอกไว้ในลิสต์ข้อมูลแล้ว ซึ่งตัวเลขนี้จะใช้ช่วยในการค้นหาสถานที่ที่ผู้ใช้ไม่แน่ใจว่าอยู่ตรงช่วงใด โดยจะดูเปรียบเทียบกับแผนที่ ที่จะกล่าวถึงในส่วนต่อไป นอกจากนี้ข้อมูลตัวเลขนี้ ยังนำไปใช้ในส่วนของการใช้อัลกอริทึมในการค้นหา ในรูปแบบของ ชื่อของโหนดเริ่มต้น (Start) และชื่อของโหนดที่เป็นจุดมุ่งหมายหรือจุดโกล (Goal ) ในกรณีที่จุดเริ่มต้นกับจุดโกลเป็นจุดเดียวกันจะเป็นการเช็คว่ามีรถเมล์สายใดบ้างที่ผ่านป้ายรถเมล์นั้น นอกจากจุดเริ่มต้น และจุดปลายทางแล้วยังมีตัวเลือกวิธีการที่จะใช้ในการค้นหาข้อมูล ซึ่งก็จะมีอยู่ 2วิธี วิธี เบรดท์-เฟิรสท์ เซิรสช์ และวิธี อิเทอเรทีฟ ดีเพนดิง เซิรสช์ โดยโปรแกรมจะทำการรับค่าที่ส่งมาจากผู้ใช้ผ่านกล่องข้อมูลที่มีเมนูเลือกนี้ ในทั้ง 3 ส่วน คือรับจุดเริ่ม รับจุดโกล และส่วนรับวิธีการค้นหา เพื่อทำการส่งให้กับส่วนค้นหาข้อมูลดำเนินการค้นหาข้อมูลต่อไป

3. ส่วนที่ 3 จะอยู่ใกล้ ๆ กับในส่วน ที่ 1 คืออยู่ทางด้านล่างของส่วนที่ 1 โดยจะมีลักษณะเป็นรายละเอียดวิธีการใช้งาน เป็นหัวข้อต่างๆ ซึ่งจะมีรายละเอียดต่าง ๆ ของข้อมูลที่ปรากฏบนแผนที่

4. ในส่วนที่ 4 นี้จะเป็นส่วนแผนที่ โดยในกรณีของส่วนอินพุตข้อมูลนี้จะใช้ให้ผู้ใช้ดูว่าจุดใดที่ผู้ใช้ต้องการให้เป็นจุดเริ่มต้น และใช้ดูว่าจุดใดที่ผู้ใช้ต้องการให้เป็นจุดโกล โดยจากส่วนที่แล้วได้อธิบายแล้วว่าในส่วนแผนที่นี้จะมีตัวเลขกำกับแต่ละจุดของแผนที่ไว้ตามรูปที่ 6 โดยจะช่วยผู้ใช้อย่างมากในการดูแผนที่เพื่อใช้ในการอินพุตข้อมูล จากเดิมเรากำหนดขอบเขตป้ายรถเมล์ไว้ 28 ป้าย ปัจจุบันมีการเพิ่มเป็น 63 ป้าย เพื่อให้สามารถขยายขอบเขตในการเซิรสช์ให้มากยิ่งขึ้นกว่าเดิมที่มีใช้อยู่ นอกจากนี้เมื่อเราได้ทำการใช้เมาส์คลิ้กเข้าไปในแผนที่แล้วก็จะเป็นการขยายแผนที่ให้มีขนาดที่ใหญ่ขึ้นเพื่อที่จะทำให้สะดวกในการดูรายละเอียดรอบ ๆ แผนที่ ซึ่งแผนที่ขนาดใหญ่จะไม่สามารถให้รายละเอียดของแผนที่ได้มาก โดยเราจะดูว่าจุดที่เราต้องการกำหนดให้เป็นจุดเริ่มอยู่ที่ส่วนใดของแผนที่ และในแผนที่ที่มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขยายใหญ่นี้จะมีความสำคัญในส่วนของการแสดงผล เนื่องจากจะใช้ในการแสดงทิศทางการเดินทางของรถเมล์ด้วย โดยรายละเอียดส่วนนี้เราจะได้ทำการนำเสนอในเนื้อหาที่จะกล่าวถึงต่อไป



*รูปที่ 6-2 แสดงตัวอย่างตัวเลขกำกับจุดบนแผนที่*

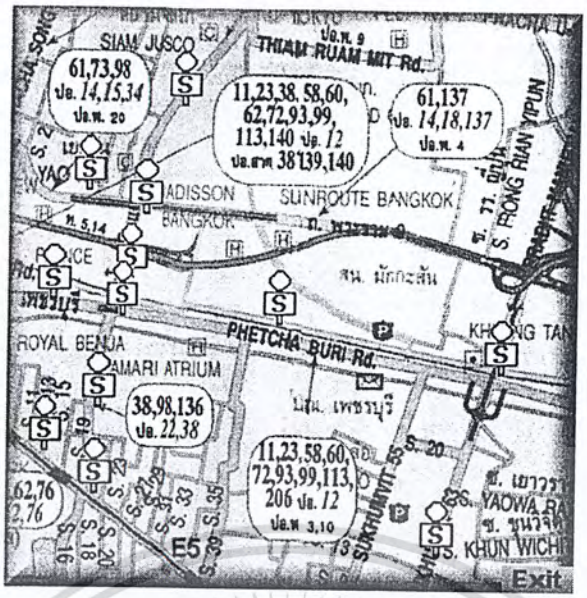
จากเนื้อหาในส่วนข้างบนที่กล่าวถึงการอินพุตข้อมูลทั้ง 4 ส่วนแล้ว เราจะพบว่าเมื่อข้อมูลที่เลือกจากส่วนที่ 2 เสร็จแล้ว จะมีการนำเอาข้อมูลตัวเลขที่กำกับไว้ไปส่งให้กับ ส่วนที่ค้นหา เพื่อทำการนำข้อมูลส่วนนี้ไปประมวลผลต่อไปในรูปของโหนดเริ่มและโหนดโกลตามที่ได้กล่าวมาแล้ว โดยข้อมูลที่ส่งไปนั้นจะอยู่ในรูปของชุดข้อมูลที่ส่งไปเป็น โหนดที่เลือกให้เป็นจุดเริ่ม โหนดที่เลือกให้เป็นจุดโกล และวิธีการในการค้นหา

**6.1.2 การอินพุตข้อมูลผ่านแผนที่**

ในการอินพุตข้อมูลผ่านตัวแผนที่ใหญ่นั้น จะเป็นการเลือกเข้าไปในตัวแผนที่ใหญ่ให้ขยายขนาดขึ้น แล้วเราจะสามารถทำการคลิ๊กเลือกเข้าไปในป้ายต่างๆ บนแผนที่ ซึ่งจะทำให้เกิดข้อความขึ้นบอกว่าป้ายดังกล่าวมีชื่อว่าอย่างไร เมื่อเราคลิ๊กอีกทีก็จะเป็นการเลือกเข้าไปตกลงว่าเป็นจุดเริ่มต้น แล้วเช่นกันกับจุดหมายปลายทางก็ทำเช่นเดียวกัน

เมื่อเราอินพุตข้อมูลป้ายรถเมล์ในแบบดังกล่าวแล้ว เราจะทำการเลือกวิธีการค้นหาตามแบบแรกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 6-3 แสดงตัวอย่างป้ายรถเมล์ที่ให้อินพุตบนแผนที่*

โดยสรุปแล้วในส่วนของการรับอินพุตข้อมูลนั้นจะมีแบ่งเป็น 4 ส่วนในการรับด้านอินพุตบนตามที่ได้กล่าวมาแล้วนั้นคือมี ส่วนแรก ส่วนการเลือกไปยังเพจอื่น ส่วนที่ 2 คือส่วนของการรับอินพุตวิธีการค้นหา และป้ายที่จะค้นหา โดยจะมีขอบเขตของป้ายที่จะสนใจเป็นจำนวน 63 ป้าย ในการอินพุตข้อมูลส่วนที่ 3 ซึ่งถือเป็นส่วนที่เพิ่มเติมออกมาจากส่วนค้นหาปกติทั่วไป คือส่วนของการพบอกวิธีการใช้ และส่วนที่ 4 จะเป็นส่วนของแผนที่ที่ใช้ในการดูข้อมูล โดยในทั้ง 4 ส่วนที่กล่าวมาในข้างต้นนี้จะต้องมีความสัมพันธ์กันทั้งหมด ยกเว้นในส่วนแรก เนื่องจากก่อนจะทำการค้นหาข้อมูลใด ๆ เราจะต้องดูข้อมูลที่ประกอบซึ่งก็คือในส่วนที่ 4 ส่วนแผนที่ เพื่อนำข้อมูลที่เราต้องการค้นไปทำการกรอกในส่วนที่ 2 ซึ่งเป็นส่วนรับอินพุตข้อมูลเข้าไป ดังที่กล่าวจะพบว่าแต่ละส่วนของการอินพุตข้อมูลจะมีความสัมพันธ์ต่อเนื่องกันในการทำงาน

**6.2 ส่วนการแสดงผลข้อมูล**

ส่วนการแสดงผลนี้เป็นส่วนที่ใช้ในการให้ข้อมูลที่ได้รับมาจากส่วนค้นหาให้ผู้ใช้ดู โดยข้อมูลที่ได้รับจากการแสดงผลจะอยู่ในรูปของโหนดที่เป็นจุดเริ่มต้น โหนดที่ผ่าน สายรถเมล์ที่ผ่าน โหนดเปลี่ยนรถ สายรถเมล์ที่จะเปลี่ยน ไป และโหนดโกล

โดยข้อมูลที่ให้ออกมานั้นจะถูกส่งไปเข้ากับข้อมูลที่ใช้แสดงผลเพื่อบอกว่าโหนดที่ให้มานั้นจะมีการแสดงผลอยู่ที่บนจุดใดในแผนที่ การแสดงผลข้อมูลสามารถแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ

**6.2.1 ส่วนการแสดงผลบนแผนที่ใหญ่**

ส่วนการแสดงผลบนแผนที่ใหญ่ โดยส่วนการแสดงผลบนแผนที่ใหญ่ นั้นก็คือจะใช้แผนที่ตามรูปแบบแผนที่ที่ใช้ในการดูข้อมูลเพื่อที่จะทำในส่วนของการอินพุตข้อมูล โดยใช้แผนที่เดียวกับส่วนนี้ แผนที่ใหญ่นั้นจะเป็นแผนที่ที่มีความมากของเนื้อหาข้อมูลแต่จะมีความด้อยของรายละเอียดโดยจะมีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดจุดขึ้นมาแทนป้ายรถเมล์แต่ละป้าย โดยในป้ายรถเมล์แต่ละป้ายจะแทนช่วงถนนที่อยู่ระหว่างแยกแต่ละแยก และแยกในทุก ๆ แยกก็จะมีเป็นโหนดกำกับไว้ในแต่ละแยก สรุปคือในแต่ละแยกและช่วงถนนบนแผนที่นั้นจะมีจุดโหนดกำกับ แทนป้ายรถเมล์แต่ละจุดตามรูป

*รูปที่ 6-4 แสดงตัวอย่างของการแสดงผลข้อมูล*
*บนแผนที่ขนาดใหญ่*

**6.2.2 ส่วนการแสดงผลบนแผนที่ขนาดขยายใหญ่**

ส่วนการแสดงผลบนแผนที่ขนาดขยายใหญ่ เมื่อเราใช้เมาส์คลิ้กเลือกไปยังส่วนการแสดงผลบนแผนที่ใหญ่ ก็จะพบว่ามีการเปลี่ยนรูปออกมาเป็นแผนที่ที่มีการขยายขนาดขึ้นมาให้มีรายละอียดมากขึ้น โดยจะมีการกำหนดจุดขึ้นมาแทนป้ายรถเมล์แต่ละป้ายตามในแผนที่ใหญ่ กล่าวคือในป้ายรถเมล์แต่ละป้ายนั้นจะแทนช่วงถนนที่อยู่ระหว่างแยกแต่ละแยก และแยกในทุก ๆ แยกก็จะมีป้ายรถเมล์กำกับด้วย สรุปคือจะมีจุดโหนดกำกับ แทนป้ายรถเมล์แต่ละจุดตามที่ได้กล่าวมา

เมื่อเรากล่าวถึงวิธีการแสดงผลแบบต่าง ๆ แล้วต่อไปเราจะมาพูดถึงหลักการที่ใช้ในการแสดงผลซึ่งจะมีขั้นตอนต่าง ๆ ตามบทที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7
# หลักการแสดงผล

จากที่ส่วนการแสดงผลนี้เป็นส่วนที่ใช้ในการให้ข้อมูลที่ได้รับมาจากส่วนค้นหาให้ผู้ใช้ดู โดยข้อมูลที่ได้รับจากการแสดงผลจะอยู่ในรูปของโหนดที่เป็นจุดเริ่มต้น โหนดที่ผ่าน สายรถเมล์ที่ผ่าน โหนดเปลี่ยนรถ สายรถเมล์ที่จะเปลี่ยน ไป และโหนดโกล โดยข้อมูลที่ให้ออกมานั้นจะถูกส่งไปเข้ากับข้อมูลที่ใช้แสดงผลเพื่อบอกว่าโหนดที่ให้มานั้นจะมีการแสดงผลอยู่ที่บนจุดใดในแผนที่ โดยข้อมูลตัวอย่างจะเป็นดังตาราง

| ชื่อโหนด | จุดแสดงผลบนแผนที่ใหญ่ | จุดแสดงผลบนรูปภาพที่ขยายแล้ว | รูปที่แสดงเมื่อขยายใหญ่ |
|---|---|---|---|
| 1 | X=639, Y=159 | X=232 , Y=101 | 641 |
| 2 | X=676 , Y=154 | X=320 , Y=98 | 641 |
| 3 | X=686 , Y=168 | X=380 , Y=105 | 641 |
| . | . | . | . |
| 28 | X=712 , Y=359 | X=416 , Y=186 | 651 |
| 29 | X=462 , Y=376 | X=556 , Y=222 | 451 |
| 30 | X=450 , Y=257 | X=535 , Y=313 | 441 |
| . | . | . | . |
| 61 | X=726 , Y=577 | X=360 , Y=303 | 661 |
| 62 | X=501 , Y=561 | X=253 , Y=316 | 561 |
| 63 | X=650 , Y=588 | X=232 , Y=462 | 661 |

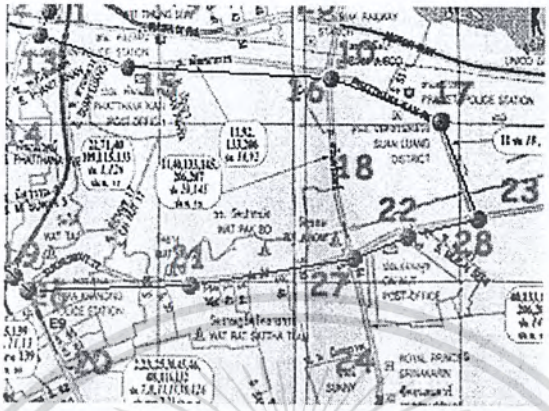
*ตารางที่ 7-1 แสดงการเก็บค่าข้อมูลของโหนดแต่ละโหนด*

จากข้อมูลที่ได้รับมาจากส่วนค้นหา ซึ่งคือหมายเลขกำกับโหนดนั้นจะถูกนำเข้าไปค้นหาข้อมูลจุดแสดงผล 2 จุดคือจุดแสดงผลบนแผนที่ใหญ่, จุดแสดงผลบนรูปภาพที่ขยายแล้ว โดยจะได้ค่า ออกมา 4 ค่า คือค่า X และค่า Y ของจุดแสดงผลบนแผนที่ใหญ่ และค่า X และค่า Y ของจุดแสดงผลบนรูปภาพที่ขยายแล้ว ซึ่งค่า X และ ค่า Y เหล่านี้จะเป็นตัวแสดงจุดเริ่มต้นของการวาดรูปโหนด และนอกจากนี้ยังมีค่ารูปที่แสดงบนแผนที่ขยายเพิ่มขึ้นมาเพื่อใช้บอกว่าโหนดดังกล่าวเมื่อเข้าไปอยู่ในแผนที่แบบขยายขนาดแล้วจะอยู่ที่แผนที่ส่วนใด ซึ่งจะได้กล่าวถึงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**7.1 ส่วนการแสดงผลบนแผนที่ใหญ่**

ส่วนการแสดงผลบนแผนที่ใหญ่ จะมีการกำหนดจุดขึ้นมาแทนป้ายรถเมล์แต่ละป้าย โดยในป้ายรถเมล์แต่ละป้ายจะแทนช่วงถนนและแยกในแต่ละแยก ซึ่งจะมีจุดโหนดกำกับ แทนป้ายรถเมล์แต่ละจุด ตามที่กำหนดให้ตามตัวอย่างการอินพุตข้อมูลในหัวข้อที่แล้ว



*รูปที่ 7-1 แสดงหน้าจอแสดงผลป้ายรถเมล์ที่ผ่าน*

เราจะแทนป้ายรถเมล์แต่ละโหนดด้วยจุดสีซึ่งจะมีความหมายต่างกันไป โดยจุดสีแต่ละจุดจะใช้ตำแหน่ง ค่า X และค่า Y ของจุดแสดงผลบนแผนที่ใหญ่ที่ได้จากตารางการแสดงผลบนแผนที่ที่ได้กล่าวมาแล้วคือตารางที่ 7-1 ซึ่งค่า X และ Y ที่นำมาใช้บนแผนที่ใหญ่นี้คือค่า X และ Y ในช่วงแรกคือแถบช่วงแรก โดยค่า X และ Y ช่วงนี้ที่นำมาใช้จะถูกนำมาเป็นพิกัดเริ่มต้นในการวาดจุดสี โดยตัวจุดสีที่ต่างกันนั้นจะให้ความหมายที่ต่างกันด้วย

โดยหากเราดูจากรูปจะพบว่า จุดสีม่วงที่ป้ายรถเมล์ 13, 15 และ 16 จะแทนป้ายรถเมล์ของรถเมล์ต่อแรกที่ขึ้น, ส่วนจุดสีส้มที่ป้ายรถเมล์ 17 จะแสดงจุดต่อรถเมล์สายแรกขึ้นสายที่สอง จุดสีแดงในโหนดหมายเลขที่ 28 นั้น จะแทนการต่อรถสายที่ 2 ขึ้นสายที่ 3 เหตุที่เป็นเช่นนี้เนื่องจาก รถเมล์สายที่ 2 มีการขึ้นเพียงช่วงโหนดเดียวจึงไม่สามารถแสดงให้เห็นได้ จึงเห็นแต่เฉพาะเพียงแค่จุดที่ใช้ต่อรถเท่านั้น และจุดสีฟ้านั้น ในป้ายรถเมล์หมายเลข 22 , 27, 21, 26 และ 19 นั้นจะแสดงป้ายรถเมล์ที่รถสายที่ 3 ที่ขึ้นวิ่งผ่าน ถ้าเราอยากรู้ว่ารถเมล์สายใดคือรถเมล์ช่วงที่ 1 ให้คลิกเข้าไปในแผนที่เพื่อขยายรูปใหญ่จะพบว่าจุดแต่ละจุดแทนรถเมล์สายใดบ้าง

ในกรณีที่มีจำนวนสายต่อมากกว่านี้จะทำให้ต้องเพิ่มจำนวนสีที่ต่างกันเพิ่มขึ้นเรื่อย ๆ โดยเรามีการกำหนดขอบเขตของสีที่ใช้ไว้ทั้งหมดจำนวน 7 สี จากการต่อรถสูงสุดได้เพียง 4 ครั้ง คือจะมี 4 สีสำหรับการต่อรถแต่ละครั้ง และอีก 3 สี สำหรับการลงจากรถมาขึ้นรถสายใหม่อีกครั้ง ซึ่งจะมีตัวบอกความหมายของจุดสีแต่ละจุดเมื่อทำการขยายแผนที่ และตัวสีแต่ละสีนี้ก็จะมีการนำมาใช้อีกครั้งเมื่อทำการขยายแผนที่แล้วด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**7.2 ส่วนการแสดงผลบนแผนที่ที่ขยายขนาด**

ในการแสดงผลของแผนที่เรานำแผนที่มาใช้ 9 ส่วนย่อย โดยจะมีลักษณะแผนที่ตามรูป โดยจะแบ่งออกเป็น 4ช่องจากรูปใหญ่ มาเป็น 1 ช่องของรูปเมื่อขยาย

โดยเราอาจจะกล่าวได้ว่าเราจะนำเอาข้อมูลแผนที่ใหญ่ซึ่งประกอบด้วยการแบ่งแผนที่ใหญ่ออกเป็นส่วน ๆ ย่อย ด้วยเส้นนั้นมาใช้ ถ้าเราสังเกตในแผนที่รูปใหญ่เราจะพบว่า มีเส้นแบ่งรูปอยู่ เราจึงนำส่วนย่อยของแผนที่ใหญ่ดังกล่าวนั้นมาทำการรวมกัน 4 ส่วนให้เป็น 1ส่วนย่อยของแผนที่เมื่อขยายขนาดแล้ว

กล่าวคือจากรูปใหญ่นั้นเราสามารถแบ่งรูปออกมาได้ขนาดเป็น 36 ส่วนเมื่อทำการขยายขนาดแล้ว คือเกิดจากแบ่งตามแนวนอนได้ 6 ส่วนย่อยคือเกิดจาก 12 ส่วนย่อยของรูปใหญ่ และแบ่งตามแนวตั้งได้ 6 ส่วนย่อยคือเกิดจาก 12 ส่วนย่อยของรูปใหญ่เช่นกัน โดยเราเขียนเป็นเลขกำกับ โดยเลขตัวหน้าคือตัวเลขลำดับในแนวนอน และตัวเลขที่สองคือตัวเลขลำดับในแนวตั้ง ส่วนตัวเลขที่ 3 นั้นจะให้มีค่าคงที่เป็น 1 เราจะได้ทั้งหมด 36 ส่วนตั้งแต่ 111- 661

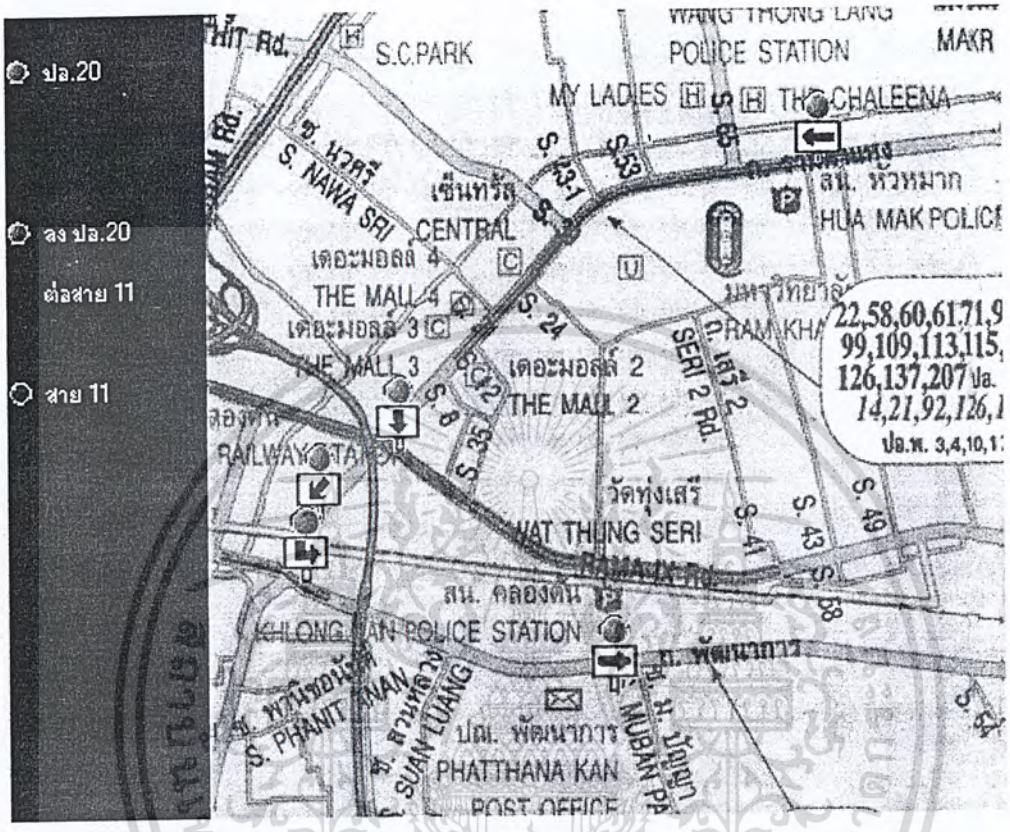
แต่เนื่องจากว่าขอบเขตของพื้นที่ที่เราทำการสนใจนั้นไม่ได้สนใจในพื้นที่ทั้งหมดของกรุงเทพ เราสนใจเฉพาะพื้นที่ในบางส่วนเท่านั้น เพราะฉะนั้นเราจึงได้ทำการตัดขอบเขตของพื้นที่ที่สนใจจากเดิม 36 ส่วน ให้เหลืออยู่เพียง 9 ส่วนเท่านั้น คือสนใจเฉพาะเพียงแค่ 9 ส่วนที่อยู่ทางด้านมุมล่างขวาของแผนที่ทั้งหมด เพราะฉะนั้น 9 ส่วนที่เราทำการสนใจจะประกอบไปด้วยส่วน 441, 451, 461, 541, 551, 561, 641, 651 และ 661 ในทั้งหมด 9 ส่วนนี้จะถูกแยกออกมาเป็นไฟล์รูปภาพอีกชุดหนึ่ง ซึ่งเราจะทำการเคลียร์เส้นที่พาดผ่านในการแบ่งรูปออกเป็น 4 ส่วนย่อย ๆ ออก แล้วเก็บไว้ เราจึงจะได้รูปภาพที่นำไปใช้เมื่อมีการขยายแผนที่ออกมา

| | | |
|---|---|---|
| 441 | 541 | 641 |
| 451 | 551 | 651 |
| 461 | 561 | 661 |

*รูปที่ 7-2 แสดงแผนที่ขนาดขยายที่เกิดจากการแบ่งรูปขนาดใหญ่มาเป็นรูปเมื่อทำการขยายใหญ่*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการขยายแผนที่เป็นรูปใหญ่นั้นจะสามารถแบ่งออกได้เป็น 2 ส่วนย่อย ๆ คือ ส่วนที่ใช้บอกสายรถเมล์และส่วนแผนที่ที่ขยายใหญ่



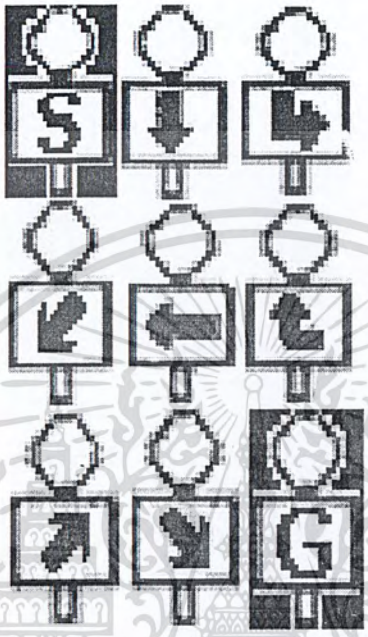

*รูปที่ 7-3 แสดงแผนที่เมื่อทำการขยายใหญ่*

จากรูปที่แสดงเราจะพบว่าเราสามารถจะแบ่งรูปออกเป็น 2 ส่วนย่อย ๆ ตามรูป ซึ่งในแต่ละส่วนย่อย ๆ นี้จะมีรายละเอียดที่ต่างกันไปดังนี้

1.ส่วนแสดงสายรถเมล์ จากที่กล่าวมาแล้วว่าในส่วนนี้จะเป็นตัวบอกว่าจุดสีแต่ละจุดจะหมายถึงรถเมล์สายใด หากมีจำนวนรถเมล์จำนวนหลายสายที่สามารถขึ้นได้ ก็จะแสดงทุกสายที่เป็นไปได้ ดังตัวอย่าง จุดสีม่วงแทน รถ ปอ. 20 ส่วนกรณีที่เป็นจุดต่อรถก็จะเขียนบอกตามรูป ว่าให้ลงจากรถสายใด ต่อรถสายใด จากตัวอย่างเป็นการลงจากรถสาย ปอ.20 ต่อสาย 11 แล้วถ้ามีการขึ้นต่อที่ 2 ก็จะบอกให้ว่าเป็นการขึ้นต่อสายใด โดยหากว่ามีการต่อสายรถเมลีจำนวนมากก็จะต้องมีการเพิ่มจำนวนข้อความตามไปด้วย โดยจะสิ้นสุดของข้อความที่เกิดขึ้นได้จำนวน 7 ข้อความ ตาทขอบเขตของการค้นหาข้อมูลที่ให้มากสุดได้ 4 ครั้ง จึงแสดงผลข้อมูลสายรถเมลีทั้งหมด 4 ครั้ง และข้อมูลที่ใช้บอกว่าเป็นการต่อรถเมล์ในรูปแบบใด ลงและขึ้นต่อสายใด อีกทั้งหมด 3 ครั้ง รวมเป็นข้อความที่เกี่ยวกับการต่อรถเมล์ ทั้งหมดรวม 7 แบบของข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.ส่วนการแสดงแผนที่ขยายใหญ่ ในส่วนนี้จะมีป้ายรถเมล์ที่มีลักษณะ แตกต่างกันไปตามทิศทางของรถที่วิ่ง และมีส่วนของรายละเอียดของแผนที่มากยิ่งขึ้น โดยจะมีรูปแบบของป้ายทิศทางต่างกันไปเช่น บอกว่าเป็นจุดเริ่ม บอกให้ไปทางขวา บอหให้ตรงขึ้นไป บอกให้เลี้ยวขวา โดยมีตัวอย่างป้ายรถเมล์ ดังรูป




*รูปที่ 7-4 รูปแสดงตัวอย่างป้ายรถเมล์แบบต่าง ๆ ที่ใช้ในแผนที่*

จากตัวอย่างรูป เราจะมีสัญลักษณ์ต่าง ๆ กันไป ซึ่งสัญลักษณ์แต่ละตัวที่แสดงตามรูป มีชื่อเรียกเรียงกันดังนี้ Start, Down, DownRight, NESW, Left, LeftUp, SWNE, SENW, Goal ซึ่งแต่ละตัวจะมีความหมายทิศทางตามชื่อ เช่น DownRight คือวิ่งลงแล้วเลี้ยวขวา, Right คือวิ่งไปทางขวา เป็นต้น

โดยลักษณะของป้ายรถเมล์จะสามารถแบ่งออกได้เป็น 4 ประเภท

1.ป้ายบอกจุดเริ่มต้น และ จุดโกล จะใช้ตัวอักษรตัว S และ G เป็นสัญลักษณ์

2.ป้ายบอกทิศทาง ตามแนวนอนและแนวดิ่ง เช่นบอกให้ตรงไป ทางซ้าย

3.ป้ายบอกทิศทาง ตามแนวทแยงมุม เหมือนป้ายในข้อ 2 แต่จะเป็นตรงไป ในเส้นทางทแยงมุม ป้ายแบบที่2 และ 3 สามารถใช้กับแยกได้ด้วย

4.ป้ายบอกทิศทางการเลี้ยว จะใช้กับจุดที่เป็นแยกเท่านั้น

โดยในส่วนของป้ายรถเมล์นั้นการกำหนดว่าจะใช้ป้ายแบบใด จะขึ้นอยู่กับ ทิศทางของโหนดที่จะไป โดยขึ้นอยู่กับ โหนดที่รับมา โหนดก่อนหน้า และโหนดถัดไป ว่าเป็นโหนดใด ( ป้ายรถเมล์ใด ) ซึ่งจะสามารถทำให้รู้ทิศทางที่โหนดนั้นต้องไปได้ โดยเราจะแบ่งรูปแบบของโหนดแต่ละโหนดออกเป็น 4 ส่วน โดยในแต่ละส่วนนั้นจะอาศัยความเป็นลักษณะป้ายในแต่ละรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.โหนดแทนป้ายทางตรงที่สิ้นสุดแผนที่ โหนดดังกล่าวนี้จะเป็นโหนดที่สิ้นสุดของแผนที่ที่เรากำหนดขอบเขตไว้ ดังนั้นตัวที่แทนโหนดได้จะเป็นได้แค่ S หรือ G เท่านั้น คือเป็นได้แค่จุดเริ่มต้น หรือจุดหมายปลายทาง เนื่องจากโหนดดังกล่าวจะไม่มีทางที่อาศัยโหนดดังกล่าวเป็นทางผ่าน ดังนั้นจึงจะไม่มีการบอกทิศทางที่จะไปต่อโดยจะถือว่าเป็นการสิ้นสุดของแผนที่แล้ว

2.โหนดประเภทที่ 2 คือโหนดที่แทนป้ายรถเมล์ที่เป็นช่วงถนน โดยโหนดดังกล่าวนี้จะแทนตัวป้ายรถเมล์ที่เป็นจุดบนถนนระหว่างแยกต่าง ๆ ซึ่งโหนดดังกล่าวจะเป็นไปได้ทั้งป้ายบอกจุดเริ่มต้นและจุดหมายปลายทาง เป็นป้ายบอกทิศทาง ตามแนวนอนและแนวดิ่ง รวมทั้งเป็นป้ายบอกทิศทางตามแนวเส้นทแยงมุม โดยเราจะแบ่งประเภทของโหนดชนิดนี้ได้เป็นอีก 4 ประเภทตามลักษณะของถนนช่วงนั้น ๆ

2.1 โหนดแทนช่วงถนนตามแนวดิ่ง จะเป็นโหนดที่เกิดขึ้นบนถนนตามแนวดิ่งหรือใกล้เคียงกับแนวดิ่ง โดยจะพิจารณาจากแผนที่ที่มีว่า มีลักษณะถนนช่วงดังกล่าวเป็นเช่นไร โดยโอกาสที่ป้ายรถเมล์จะเกิดขึ้นบนถนนแบบนี้นั้น จะเป็นไปได้ 2 กรณีคือ เป็นป้ายรถเมล์ที่แสดงว่ารถเมล์วิ่งจากบนลงล่าง และวิ่งจากด้านล่างขึ้นไปด้านบน โดยป้ายรถเมล์ทั้ง 2 รูปแบบดังกล่าวนั้นจะเกิดขึ้นบนถนนแบบเดียวกัน แต่ต่างกันตรงที่เป็นขาไป และขากลับเท่านั้น

2.2 โหนดแทนช่วงถนนแนวนอน จะเป็นโหนดที่เกิดขึ้นบนถนนตามแนวนอน หรือมีลักษณะที่ใกล้เคียงกับถนนแนวนอน หรือมีบางช่วงของถนนเป็นแนวนอน โดยจะพิจารณาจากแผนที่ที่มีว่า มีลักษณะถนนช่วงดังกล่าวเป็นเช่นไร โดยโอกาสที่ป้ายรถเมล์จะเกิดขึ้นบนถนนแบบนี้นั้น จะเป็นไปได้ 2 กรณีคือ เป็นป้ายรถเมล์ที่แสดงว่ารถเมล์วิ่งจากซ้ายไปขวา และวิ่งจากด้านขวาขึ้นไปด้านซ้าย โดยป้ายรถเมล์ทั้ง 2 รูปแบบดังกล่าวนั้นจะเกิดขึ้นบนถนนแบบเดียวกัน แต่ต่างกันตรงที่เป็นขาไป และขากลับเท่านั้น

2.3 โหนดแทนช่วงถนนแนวทแยงมุมจากด้านบนซ้ายไปด้านล่างขวา และทิศทางจากด้านบนขวาไปด้านล่างซ้าย โดยจะเป็นโหนดที่เกิดขึ้นบนถนนตามแนวทแยงมุมด้านบนซ้ายไปล่างขวา และล่างขวาไปบนซ้าย หรือมีลักษณะที่ใกล้เคียงกับถนนแนวทแยงมุมดังกล่าว หรือมีบางช่วงของถนนเป็นแนวทแยงมุมดังกล่าว โดยจะพิจารณาจากแผนที่ที่มีว่า มีลักษณะถนนช่วงดังกล่าวเป็นเช่นไร โดยโอกาสที่ป้ายรถเมล์จะเกิดขึ้นบนถนนแบบนี้นั้น จะเป็นไปได้ 2 กรณีคือ เป็นป้ายรถเมล์ที่แสดงว่ารถเมล์วิ่งจากด้านบนซ้ายไปด้านล่างขวา และวิ่งจากด้านล่างขวาขึ้นไปด้านบนซ้าย โดยป้ายรถเมล์ทั้ง 2 รูปแบบดังกล่าวนั้นจะเกิดขึ้นบนถนนแบบเดียวกัน แต่ต่างกันตรงที่เป็นขาไป และขากลับเท่านั้น

2.4 ป้ายรถเมล์แทนช่วงถนนช่วงตรงแบบที่ 4 จะคล้ายกับแบบที่ 3 แต่จะทแยงมุมคนละด้านคือโหนดแทนช่วงถนนแนวทแยงมุมจากด้านบนขวาไปด้านล่างซ้าย และทิศทางจากด้านบนซ้ายไปด้านล่างขวา โดยจะเป็นโหนดที่เกิดขึ้นบนถนนตามแนวทแยงมุมด้านบนขวาไปล่างซ้าย และล่างซ้ายไปบนขวา หรือมีลักษณะที่ใกล้เคียงกับถนนแนวทแยงมุมดังกล่าว หรือมีบางช่วงของถนนเป็นแนวทแยงมุมดังกล่าว โดยจะพิจารณาจากแผนที่ที่มีว่า มีลักษณะถนนช่วงดังกล่าวเป็นเช่นไร โดยโอกาสที่ป้ายรถเมล์จะเกิดขึ้นบนถนนแบบนี้นั้น จะเป็นไปได้ 2 กรณีคือ เป็นป้ายรถเมล์ที่แสดงว่ารถเมล์วิ่งจากด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ล่างซ้ายไปด้านบนขวา และวิ่งจากด้านบนขวาขึ้นไปด้านล่างซ้าย โดยป้ายรถเมล์ทั้ง 2 รูปแบบดังกล่าวนั้นจะเกิดขึ้นบนถนนแบบเดียวกัน แต่ต่างกันตรงที่เป็นขาไป และขากลับเท่านั้น

3. โหนดประเภทที่ 3 คือโหนดที่แทนป้ายรถเมล์ที่เป็นแยก คือโหนดที่แทนแยกต่าง ๆ ที่ถนนแต่ละเส้นมาตัดกัน หรือมาสิ้นสุด โดยที่เป็นเสมือนจุดเชื่อมระหว่างถนนเส้นตรงต่าง ๆ กับถนนอีกช่วงหนึ่ง โดยโหนดที่เป็นแยกที่มีจะมีอยู่ 3 ประเภท คือ

3.1 โหนดที่เป็น 4 แยก จะเป็นโหนดที่มีให้เห็นทั่วไป ในแผนที่ โหนดดังกล่าวเกิดจากการที่มีถนนเส้นหนึ่งมาตัดถนนอีกเส้นทำให้เกิดเป็น 4 แยก หรืออาจจะเกิดจากการที่มี สาม แยก 2 สามแยกอยู่ใกล้เคียงกันมากจนสามารถยุบรวมมาเป็น 4 แยกได้ โดยโอกาสของป้ายที่จะเกิดขึ้นนั้นจะเป็น ป้ายในลักษณะของป้ายบอกทิศทางตามแนวดิ่งทิศทางตามแนวนอน และทิศทางการเลี้ยวโดยที่ จะขึ้นอยู่กับโหนดที่เกิดขึ้น โดยจะเปรียบเทียบให้มีสี่แยกอยู่ตรงกลางและถนนที่วางอยู่ตามแนวดิ่งจะแบ่งออกเป็น 2ช่วง คือ ช่วงบน และช่วงล่าง ส่วนที่อยู่ตามแนวนอนจะแบ่งออกเป็นช่วงซ้าย และช่วงขวาของแยก โดยให้จุดตัดกันของถนนทั้ง 2 เส้นเป็น 4 แยก โดยป้ายรถเมล์แสดงทิศทางของโหนดที่จะเกิดขึ้นได้จะมีอยู่ 12 แบบ ดังนี้

3.1.1 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางขวาแล้วเลี้ยวขึ้น (RightUp) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางซ้ายของแยกไปยังถนนที่อยู่ด้านบนของแยก

3.1.2 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางขวาแล้วเลี้ยวลง (RightDown) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางซ้ายของแยกไปยังถนนที่อยู่ด้านล่างของแยก

3.1.3 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางซ้ายแล้วเลี้ยวขึ้น (LeftUp) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางขวาของแยกไปยังถนนที่อยู่ด้านบนของแยก

3.1.4 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางขวาแล้วเลี้ยวลง (LeftDown) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางขวาของแยกไปยังถนนที่อยู่ด้านล่างของแยก

3.1.5 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางด้านล่างแล้วเลี้ยวซ้าย (DownLeft) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านบนของแยกไปยังถนนที่อยู่ด้านซ้ายของแยก

3.1.6 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางด้านล่างแล้วเลี้ยวขวา (DownRight) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านบนของแยกไปยังถนนที่อยู่ด้านขวาของแยก

3.1.7 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงขึ้นไปทางด้านบนแล้วเลี้ยวซ้าย (UpLeft) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านล่างของแยกไปยังถนนที่อยู่ด้านซ้ายของแยก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.8 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางด้านบนแล้วเลี้ยวขวา (UpRight) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านล่างของแยกไปยังถนนที่อยู่ด้านขวาของแยก

3.1.9 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงลงล่าง (Down) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านบนของแยกไปยังถนนที่อยู่ด้านล่างของแยก

3.1.10 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงขึ้นบน (Up) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านล่างของแยกไปยังถนนที่อยู่ด้านบนของแยก

3.1.11 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางซ้าย (Left) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านขวาของแยกไปยังถนนที่อยู่ด้านซ้ายของแยก

3.1.12 ป้ายรถเมล์ที่แสดงว่ารถจะวิ่งตรงไปทางขวา (Right) ถ้ามองจากสี่แยกแล้วโหนดนี้จะเกิดขึ้นได้ก็ต่อเมื่อเป็นการเคลื่อนที่จากถนนทางด้านซ้ายของแยกไปยังถนนที่อยู่ด้านขวาของแยก

ป้ายรถเมล์ทั้ง 12 แบบดังกล่าวมานี้จะนำไปใช้ในส่วนของ ป้ายรถเมล์ที่แสดงทิศทางของโหนด 3 แยก และโหนด 5 แยกด้วย ซึ่งจะอธิบายต่อไป

3.2โหนดที่เป็น 3 แยก จะเป็นโหนดที่มีถนนเส้นหนึ่งมาสิ้นสุดที่ถนนอีกเส้นหนึ่ง โดยจากแผนที่อาจจะเป็นถนน 4 แยกที่มีช่วงหนึ่งของถนนที่ไม่มีรถเมล์สายใดผ่าน จึงไม่ได้นำมาคิด จึงทำให้เหลือแยกที่นำมาคิดเพียง 3 แยก โหนดที่เป็น 3 แยกจะมีลักษณะป้ายรถเมล์ที่เกิดขึ้นได้เหมือนกับ 4 แยก แต่ จะไม่คิดในแยกที่ไม่ได้ใช้เพราะฉะนั้นโหนดที่เกิดขึ้นมาจะเป็นไปได้ 4 แบบคือ

3.2.1 โหนดที่เป็น 3 แยก แบบที่ไม่มีโหนดด้านขวา คือเป็น 3 แยก ที่มีเพียงด้านบน ด้านล่างและมีถนนด้านซ้ายมาบรรจบที่แยก ป้ายรถเมล์ที่เกิดขึ้นได้ในถนนแบบดังกล่าวคือ ป้ายรถเมล์ในแบบที่ 3.1.1, 3.1.2, 3.1.5, 3.1.7 , 3.1.9, 3.1.10 รายละเอียดตามหัวข้อของโหนดที่เป็น 4 แยก

3.2.2 โหนดที่เป็น 3 แยก แบบที่ไม่มีโหนดด้านซ้าย คือเป็น 3 แยก ที่มีเพียงด้านบน ด้านล่างและมีถนนด้านขวามาบรรจบที่แยก ป้ายรถเมล์ที่เกิดขึ้นได้ในถนนแบบดังกล่าวคือ ป้ายรถเมล์ในแบบที่ 3.1.3, 3.1.4, 3.1.6, 3.1.8, 3.1.9, 3.1.10 รายละเอียดตามหัวข้อของโหนดที่เป็น 4 แยก

3.2.3 โหนดที่เป็น 3 แยก แบบที่ไม่มีโหนดด้านบน คือเป็น 3 แยก ที่มีเพียงด้านซ้าย ด้านขวาและมีถนนด้านล่างขึ้นมาบรรจบที่แยก ป้ายรถเมล์ที่เกิดขึ้นได้ในถนนแบบดังกล่าวคือ ป้ายรถเมล์ในแบบที่ 3.1.2, 3.1.4, 3.1.7, .3.1.8, 3.1.11, 3.1.12 รายละเอียดตามหัวข้อของโหนดที่เป็น 4 แยก

3.2.4 โหนดที่เป็น 3 แยก แบบที่ไม่มีโหนดด้านบน คือเป็น 3 แยก ที่มีเพียงด้านซ้าย ด้านขวาและมีถนนด้านบนลงมาบรรจบที่แยก ป้ายรถเมล์ที่เกิดขึ้นได้ในถนนแบบดังกล่าวคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้ายรถเมล์ในแบบที่ 3.1.1, 3.1.3, 3.1.4, .3.1.5, 3.1.11, 3.1.12 รายละเอียดตามหัวข้อของโหนดที่เป็น 4 แยก

3.3 โหนดที่เป็น 5 แยก เป็นโหนดที่มีความยากในการจัดโหนดให้อยู่ในรูปแบบที่แสดงผลได้โดยง่าย เนื่องจากโหนดดังกล่าวจะมีลักษณะของถนนที่ผิดแปลกออกไปจาก แยกทั่วๆ ไป คืออาจจะเป็นการรวมของแยกใกล้ ๆ กัน หลายแยกจนมีลักษณะคล้าย 5 แยก หรืออาจจะเป็น 5 แยกจริง ๆก็ได้ แต่ในสโคปที่เราสนใจ 5 แยก จะเกิดจากการรวมตัวกันของแยกต่างๆ การกำหนดทิศทางป้ายรถเมล์จะต้องมีกุรกำหนดที่แน่นอน และเนื่องจากโหนดที่เป็น 5 แยกมีจำนวนน้อยการแสดงผลโดยการแทนค่าโดยดูจากแผนที่ จึงพอจะทำได้ การแสดงผลป้ายรถเมล์จะใช้ป้ายรถเมล์แบบแสดงการเลี้ยวเช่นเดียวกับแบบ 4 แยก และแบบ 3 แยก แต่ยังไม่มีอัลกอริทึมที่เหมาะสมในการแสดงผลแบบ 5 แยกเหมือนกับใน 2 แบบแรก

**7.3 การนำข้อมูลมาแสดงผลทิศทาง**

จากที่กล่าวแล้วในหัวข้อที่แล้วว่าเราจะแบ่งโหนดออกเป็น 3 ส่วน คือโหนดส่วนที่เป็น จุดสิ้นสุดแผนที่ โหนดที่แทนช่วงถนน และโหนดที่แทนแยก หลักการแสดงผลจะเป็นการนำข้อมูลโหนดแต่ละโหนดที่มีการเก็บไว้ว่าเป็นโหนดชนิดใด โดยเราจะแบ่งเป็นโหนด 4 ประเภท

1. โหนดส่วนที่เป็นจุดสิ้นสุดแผนที่
2. โหนดที่แทนช่วงถนน
3. โหนดที่แทน 3 แยก และ 4 แยก
4. โหนดที่แทน 5 แยก

เมื่อเราได้โหนด 4 แบบนี้แล้ว เราจะนำข้อมูลโหนดแต่ละโหนดตั้งแต่ 1 –63 ในสโคปที่เราสนใจใส่ลงไปในแต่ละประเภทของโหนดเพื่อแบ่งแยก เช่นโหนด 1 เป็นโหนดส่วนที่สิ้นสุดของแผนที่ ก็ใส่ในส่วนที่ 1 โหนด 2 ก็เป็นโหนดส่วนที่สิ้นสุดของแผนที่ ก็ใส่ในส่วนที่ 1 ทำเช่นนี้ไปจนถึงโหนด 63 เราจะได้โหนดแต่ละประเภทแบ่งแยกเอาไว้เป็นส่วน ๆ เพื่อใช้ในการแสดงผลต่อไป

เมื่อเราได้ข้อมูลออกมาจากส่วนการค้นหาข้อมูล เราจะนำข้อมูลที่ได้ซึ่งจะอยู่ในรูปแบบของอาร์เรย์ของข้อมูลนั้น ไปเข้าสู่อัลกอริทึมในการแสดงผล โดยเริ่มจากข้อมูลตัวแรกเราจะใส่ให้เป็นโหนดสตาร์ท ก่อน จากนั้นเราจะนำเอาข้อมูลในตัวที่สองของอาเรย์ไปตรวจสอบว่าอยู่เป็นโหนดแบบใดก็จะแสดงผลแบบนั้น ทำเช่นนี้ไปเรื่อย ๆ จนถึงโหนดตัวสุดท้ายของชุด อาเรย์ ก็จะใส่เป็นโหนดโกล โดยจะมีหลักการแสดงผลโดยอาศัยเนื้อหาการแสดงผลทิศทางแบบต่างๆซึ่งบางส่วนที่ได้กล่าวไปแล้วในหัวข้อก่อน โดยอาจสรุปได้ดังนี้

**7.3.1 แสดงผลทิศทางส่วนที่เป็นจุดสิ้นสุดของแผนที่**

โดยการจะเป็นโหนดประเภทนี้ได้นั้นในชุดข้อมูลอาเรย์จะต้องเป็นตัวแรกหรือไม่ก็ตัวสุดท้ายของชุดข้อมูล เนื่องจากจะไม่มีโหนดใดที่เชื่อมต่อออกไปจากโหนดประเภทนี้ดังนั้น ถ้าเราใส่ข้อมูลไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วพบว่าเป็นโหนดประเภทนี้ ก็ให้ดูว่าเป็นโหนดตัวแรก หรือตัวสุดท้าย ก็จะสามารถใส่ข้อมูลได้ตรงตามต้องการ ว่าป้ายทิศทางที่แสดงนั้น จะเป็นตัว "S" ซึ่งหมายถึง Start หรือจุดเริ่มต้น หรือจะเป็นตัว "G" ซึ่งหมายถึง Goal หรือจุดสิ้นสุด

**7.3.2 แสดงผลทิศทางในช่วงถนน**

การแสดงผลทิศทางช่วงนี้จะต้องมีการกำหนดว่าโหนดแต่ละโหนดที่แสดงนั้นจะมีทิศทางเป็นอย่างไรทั้งขาไปและขากลับเก็บไว้เป็นข้อมูลชุดหนึ่งเพื่อใช้แสดงผล โดยตัวอย่างเช่น โหนด 6 ซึ่งมีลักษณะถนนเป็นแนวดิ่ง ก็จะมีการเก็บข้อมูลว่าโหนด 6 นั้นสามารถเป็นไปได้ 2 ทางคือ Up ทิศทางจากล่างขึ้น บน และ Down ทิศทางจากบนลงล่าง โดยเราจะเก็บแยกไว้เป็น 2 ส่วน ซึ่งข้อมูลชุดแรกของแต่ละโหนดจะเก็บทิศทาง ที่มีลักษณะ ขึ้น ไปทางซ้าย ล่างขวาไปบนซ้าย และจากล่างซ้ายไปบนขวา ไว้ ส่วนชุดสองนั้นจะเก็บ ทิศทางที่มีลักษณะทิศทางตรงข้ามกับชุดที่ 1 ได้แก่ ลงล่าง ไปทางขวา จากบนขวาไปล่างซ้าย และจากบนซ้ายไปล่างขวาด้วย เมื่อเราเก็บทิศทางไว้แล้วเราก็จะนำไปเปรียบเทียบกับ การแสดงผลทิศทางที่ได้จากแยก

**7.3.3 แสดงผลทิศทางช่วง 3 แยกและ 4 แยก**

การแสดงผลทิศทางในช่วงแยกนั้นจะมีการเก็บลักษณะของทิศทางออกเป็น 5 อาร์เรย์ โดยจะแบ่งเป็น อาร์เรย์ทั้งหมด 5 ทิศทางดังนี้

1. อาร์เรย์เก็บชื่อโหนดของ 4 แยกและ 3 แยก
2. อาร์เรย์เก็บชื่อโหนดที่อยู่ทิศเหนือ (N) ของแยก
3. อาร์เรย์เก็บชื่อโหนดที่อยู่ทิศตะวันออก (E) ของแยก
4. อาร์เรย์เก็บชื่อโหนดที่อยู่ทิศใต้ (S) ของแยก
5. อาร์เรย์เก็บชื่อโหนดที่อยู่ทิศตะวันตก (W) ของแยก

โดยถ้าเป็น 4 แยกเมื่อโหนดใดเป็นโหนดที่อยู่ตรงแยกใด เราจะนำเอาโหนดที่อยู่ทางทิศทางต่าง ๆ ของแยกนั้นมาใส่ลงไปที่อาร์เรย์แต่ละตัวตามทิศทางที่กำหนดไว้ แต่ถ้าเป็น 3 แยก เราจะกำหนดให้โหนดที่เป็นทางตันเป็นโหนด 0 ซึ่งไม่มีค่าใด ๆ เพื่อเป็นการให้การแสดงผลทิศทางแบบ 3 แยกสามารถใช้การแสดงผลได้เหมือนการแสดงผล ของ แบบ 4 แยก ซึ่งเมื่อเราได้อาเรย์ทั้งหมด ซึ่งมีการกำหนดค่าไว้แล้วในข้อมูลที่ได้กำหนดไว้แล้ว

ขณะที่เราได้ข้อมูลว่าโหนดที่อยู่ในผลลัพธ์ที่มาจากส่วนการค้นหานั้นเป็นโหนดที่เป็นลักษณะแยก เราก็จะนำค่าโหนดก่อนหน้า และโหนดถัดไปมาคิด ร่วมด้วย โดยโหนดก่อนหน้าจะได้จากลำดับของตำแหน่งอาร์เรย์ปัจจุบันลบ 1 และโหนดถัดไปจะได้จากลำดับของตำแหน่งอาร์เรย์ปัจจุบันบวก 1 เมื่อได้ทั้ง 2 โหนดนี้ มาแล้วเราก็จะนำมาตรวจสอบตามตารางว่า ผลลัพธ์ที่แสดงผลของโหนดที่จุดแยกนั้นควรจะเป็นเท่าไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| โหนดก่อนหน้า | โหนดถัดไป | ลักษณะสัญลักษณ์ |
|---|---|---|
| N | S | Down |
| N | E | DownRight |
| N | W | DownLeft |
| E | N | LeftUp |
| E | S | LeftDown |
| E | W | Left |
| S | N | Up |
| S | E | UpRight |
| S | W | UpLeft |
| W | N | RightUp |
| W | E | Right |
| W | S | RightDown |

*ตารางที่ 7-2 แสดงหลักการคำนวณวิธีแสดงผลของโหนดที่เป็นแยก*

จากตารางข้างบนเราจะนำค่าโหนดปัจจุบัน เทียบกับโหนดก่อนหน้า และโหนดถัดไป โดยมีหลักการในการกำหนดค่าสัญลักษณ์ต่าง ๆ ในการเขียนตารางดังนี้ โหนดที่อยู่บนแยกของถนน จะสามารถเป็นไปได้ทุกรูปแบบของสัญลักษณ์ ขึ้นอยู่กับแนวถนนหลักการเลี้ยวจุดสิ้นสุด คือ Left, Right, Up, Down, LeftUp, LeftDown, RightUp, RightDown, UpLeft, UpRight, DownLeft, DownRight รวมถึง Start และ Goal ถ้าโหนดนั้นเป็นข้อมูลชุดแรกและชุดสุดท้ายของอาร์เรย์ แต่จะถูกกรองก่อนจะเข้าส่วนแสดงผลทิศทางของโหนดแยกแล้ว

หลังจากที่เข้าตารางนี้แล้ว ผลที่ได้นอกจากจะเอาไปแสดงผลในโหนดแล้ว ยังนำไปใช้แสดงผลต่อในโหนดถัดไป ซึ่งถ้าเป็นช่วงของถนน เราก็จะนำค่าโหนดถัดไปของโหนดแยกไปใช้ กล่าวคือ ถ้าโหนดถัดไปของแยกเป็นโหนดที่ เป็นทิศใต้หรือทิศตะวันออก เราก็จะนำค่าข้อมูลชุดที่ 2 ของถนนตรงไปใช้ คือถือเป็นทิศที่1 คือ ลงล่าง ไปทางขวา จากบนขวาไปล่างซ้าย และจากบนซ้ายไปล่างขวา แต่ถ้าไม่ใช่จะนำข้อมูลชุดที่ 1 ไปใช้ ซึ่งได้แก่ ขึ้น ไปทางซ้าย ล่างขวาไปบนซ้าย และจากล่างซ้ายไปบนขวา โดยจะเห็นได้ว่าในแต่ละส่วนของการแสดงผลมีความสัมพันธ์กันทั้งหมด

### 7.3.4 แสดงผลทิศทางช่วง 5 แยก

จากที่กล่าวแล้วว่าโหนด 5 แยกจะมีการกำหนดค่าเข้าไปในแต่ละจุดของช่วงถนนที่เชื่อมกับแยก ว่าถ้าจากถนนช่วงนี้ไปยังถนนช่วงนั้นจะต้องแสดงทิศทางอย่างไร เนื่องจากยังไม่มีอัลกอริทึมที่เหมาะสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สามารถใช้ได้ทุก ๆ 5 แยก เหมือนกับของ 3 แยก และ 4 แยก ตัวอย่างการเก็บข้อมูลโหนดของโหนด 4 ซึ่งมีลักษณะเป็น 5 แยก

| โหนดหลัง<br>โหนดก่อน | โหนด 1 | โหนด 2 | โหนด 3 | โหนด 5 | โหนด 6 |
|---|---|---|---|---|---|
| โหนด 1 | X | Right | DownRight | DownLeft | RightDown |
| โหนด 2 | Left | X | DownRight | DownLeft | LeftDown |
| โหนด 3 | UpLeft | UpRight | X | Left | LeftDown |
| โหนด 5 | UpLeft | UpRight | Right | X | RightDown |
| โหนด 6 | UpLeft | UpRight | UpRight | UpLeft | X |

*ตารางที่ 7-3 แสดงหลักการคำนวณวิธีแสดงผลของโหนดที่เป็น 5 แยก*

ขณะที่เราได้ข้อมูลมาเราตรวจสอบพบว่าเป็นโหนดแบบ 5 แยก เราก็จะดูในตารางเปรียบเทียบโดยใช้หลักการเช่นเดียวกับ 4 แยกและ 3 แยก คือ จะนำโหนดก่อนหน้า และโหนดถัดไปมาคิด โดยดูจากตารางเปรียบเทียบเฉพาะโหนด ตามตัวอย่างในตารางแล้วนำค่าที่มีไปแสดงผล

การเก็บข้อมูลวิธีดังกล่าวของ 5 แยก ถือว่าเป็นที่สิ้นเปลืองเนื่องจากถ้ามีโหนดเพิ่มขึ้นก็ต้องดำเนินการสร้างตารางขึ้นมาตามจำนวนโหนด 5 แยกที่เพิ่มขึ้นนั้น แต่เนื่องจากว่า โอกาสเกิด 5 แยกมีจำนวนน้อยจึงสามารถทำวิธีนี้ได้ โดยในขอบเขตที่เราสนใจนั้นมีโอกาสเกิด 5 แยกอยู่ 2 จุด จึงมีตารางเปรียบเทียบอยู่ 2 ตาราง

จากที่กล่าวมาเป็นเนื้อหาของการติดต่อกับผู้ใช้และการแสดงผล อาจจะมีบางเนื้อหาที่ยังบกพร่อง หรือไม่สมบูรณ์ในแง่ของการใช้งานจริง โดยในการทำงานเราจะสนใจเพียงส่วนหนึ่งของแผนที่ ซึ่งขอบเขตนั้นจะกล่าวในหัวข้อถัดไป

**7.4 การแสดงผลสายรถเมล์**

ถัดจากเรื่องการแสดงผล ก็จะพูดถึงเรื่องของการประมวลผลสายรถเมล์ จากที่กำหนดไว้ได้กำหนดไว้ว่าจะให้มีการต่อสายรถเมล์ไม่เกิน 4 สาย เพราะฉะนั้น จะมีจุดแสดงบนแผนที่อยู่ทั้งหมด 7 แบบ โดยเราจะใช้สีเป็นตัวแยกแทนจุดต่าง ๆ กัน โดยสีดังกล่าวนี้จะมีใช้ทั้งในแผนที่ใหญ่และแผนที่เล็ก โดยจุดทั้ง 7 แบบมีลักษณะดังนี้

1.จุดสีม่วง แทนโหนดที่รถเมล์คันแรกวิ่งผ่าน

2.จุดสีส้ม แทนโหนดที่มีการลงจากรถเมล์คันที่หนึ่งเพื่อต่อรถเมล์คันที่สอง

3.จุดสีน้ำเงิน แทนโหนดที่รถเมล์คันที่สองวิ่งผ่าน

4.จุดสีแดง แทนโหนดที่มีการลงจากรถเมล์คันที่สองเพื่อต่อรถเมล์คันที่สาม

5.จุดสีฟ้า แทนโหนดที่รถเมล์คันที่สามวิ่งผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.จุดสีเหลือง แทนโหนดที่มีการลงจากรถเมล์คันที่สามเพื่อต่อรถเมล์คันที่สี่

7.จุดสีเขียว แทนโหนดที่รถเมล์คันที่สี่วิ่งผ่าน

จากที่เห็นทั้งหมด 7 จุด นั้นเราสามารถแบ่งออกได้เป็น 4 จุดที่เกี่ยวกับการวิ่งของรถเมล์สายที่ 1-4 และอีก 3 จุดจะแทนการต่อรถเมล์ในแต่ละครั้งทั้งหมด 3 ครั้ง จึงทำให้จำเป็นต้องใช้จุดทั้งหมด 7 จุด ในการแสดงผลดังกล่าว

โดยการแสดงผลในส่วนของการต่อสายรถเมล์นี้จะนำข้อมูลที่ได้จากส่วนการค้นหาซึ่งจะให้มาเป็นชุดข้อมูลของอาร์เรย์ซึ่งจะมีขนาดของข้อมูลแบ่งออกเป็นกลุ่มๆ โดยถ้าขนาดของชุดข้อมูลนั้นเท่ากับ 3 แสดงว่าผลลัพธ์ที่ได้จากการค้นหานั้นมี 1 สาย เพราะฉะนั้นผลลัพธ์ที่มีโอกาสเกิดขึ้นได้คือถ้าขนาดของข้อมูลเท่ากับ 6 จะมีการขึ้นรถ 2 สาย และถ้าขนาดของข้อมูลเป็น 9 จะมีการขึ้นรถ 3 สาย และโอกาสสุดท้ายที่เป็นไปได้คือขนาดของข้อมูลมีขนาดเท่ากับ 12 จะมีการต่อรถ 4 สาย ซึ่งถือว่าเป็นการต่อรถมากสุดที่เราสนใจ

จากที่บอกว่าเราแบ่งกลุ่มของผลลัพธ์ออกเป็นกลุ่ม ๆ กลุ่มละ 3 ข้อมูล ซึ่งข้อมูลดังกล่าวประกอบด้วยข้อมูล แสดงตัวสายรถเมล์ ข้อมูลโหนดที่เริ่มต้นขึ้นสายรถเมล์นั้น ข้อมูลโหนดที่สิ้นสุดที่ลงจากสายรถเมล์นั้น โดยเราจะนำค่าทั้ง 3 ไปแสดงผลยังส่วนต่าง ๆ กัน โดยถ้าเป็นการแสดงผลในแผนที่ใหญ่แล้ว จะใช้ข้อมูลเพียง 2 ตัวหลังคือต้องการทราบเพียง โหนดที่เริ่มต้น และโหนดที่สิ้นสุดเท่านั้น แต่เมื่อเราต้องการขยายดูแผนที่ขยายแล้วเราจำเป็นต้องแสดงผลว่ารถเมล์สายที่ใช้จุดสีม่วงแทนหรือเป็นรถเมล์คันที่ 1 นั้นคือสายอะไร เราจึงต้องนำข้อมูลสายรถเมล์ดังกล่าวนั้นมาเปรียบเทียบเพื่อหาชื่อของสายรถเมล์ ในตาราง เช่น 1 จะแทนสาย ปอ.1 2 จะแทนรถเมล์สาย 2 ซึ่งจะใช้บอกสายรถเมล์ทั้งหมดที่มีอยู่ว่าหมายเลขใดแทนชื่อข้อมูลสายใด

**7.5 ขอบเขตการทำงาน**

ขอบเขตการทำงานในการแสดงผล ซึ่งส่งผลต่อการค้นหาด้วยนั้น แต่เดิมได้กำหนดขอบเขตในการค้นหาไว้ทั้งกรุงเทพ แต่ต่อมาได้มีปัญหาบางประการเกิดขึ้นเนื่องจากพบว่าการมีขอบเขตของเนื้อหาที่ครอบคลุมทั่วทั้งกรุงเทพนั้นจะทำให้เกิดการดำเนินงานที่มีความล่าช้า และยังจะทำให้มีปัญหาเกิดขึ้นมากทั้งจากส่วนข้อมูลที่ต้องเพิ่มเติมมากขึ้น และส่วนอัลกอริทึมที่ต้องรองรับปัญหาต่าง ๆ ที่มีมากขึ้น ตามขนาดพื้นที่ที่เพิ่มขึ้น และยังรวมถึงปัญหาของแผนที่ เนื่องจากในช่วงตัวเมืองชั้นในนั้น จะมีขนาดที่เล็กในถนนแต่ละสาย จึงเป็นการยากที่จะดำเนินการวางโหนดแต่ละโหนดให้ผู้ใช้สังเกตได้ง่าย เนื่องจากจะวางโหนดแต่ละโหนดชิดกันเกินไป

ในขอบเขตการทำงานในช่วงแรกเราสนใจการทำงานอยู่ 4 ส่วนย่อยของแผนที่ คือโดยกินพื้นที่ทางเหนือตั้งแต่สี่แยกลำสาลี ถึงทางใต้ที่ถนนศรีนครินทร์ ช่วงซีคอน ทางตะวันออกตั้งแต่เขตประเวศ ไปจดที่สี่แยกคลองตัน แต่ต่อมาเราได้เพิ่มขอบเขตของความสนใจขยายออกมาเป็น 9 ส่วนย่อยดังที่กล่าวไว้ในตอนต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนขอบเขตการทำงานในรูปแบบแผนที่สามารถดูได้จากรูป

| | | |
|---|---|---|
| ดินแดง<br>สี่แยกอโศก<br>เอกมัย เพชรบุรี | รามคำแหง<br>พัฒนาการ<br>พระราม 9 ตัดใหม่ | สี่แยกลำสาลี<br>ลาดพร้าว<br>สี่แยกพัฒนาการ |
| คลองเตย สุขุมวิท<br>กล้วยน้ำไท<br>สุขุมวิท 71 | อ่อนนุช<br>พระโขนง<br>บางจาก | สี่แยกอ่อนนุช<br>ซีคอนสแควร์<br>ประเวศ |
| พระประแดง | สี่แยกบางนา<br>สำโรง | สี่แยกบางนา กม.4<br>อุดมสุข ม. ราม 2 |

*รูปที่ 7-5 แสดงขอบเขตการทำงานแสดงโดยแผนที่ใหญ่*

จากขอบเขตที่ได้แสดงมานั้นจะพบว่ามีการขยายขอบเขตขึ้นเพื่อทดสอบดูว่า ถ้าเราเพิ่มขอบเขตจะพบปัญหาใด ๆ เพิ่มเติมหรือไม่ แต่จากการตรวจสอบดูพบว่าปัญหาที่เกิดขึ้นมีน้อย ดังนั้นคาดว่าจะสามารถขยายข้อมูลเพิ่มเติมให้ครอบคลุมเนื้อหามากยิ่งขึ้นได้

โดยในรูปข้อมูลบางส่วนที่หายไปคือในส่วนของพระประแดงนั้น จะไม่มีการสนใจสายรถเมล์ที่วิ่งผ่านบริเวณนั้น เนื่องจากไม่สามารถจับมาต่อเนื่องกับรถเมล์ส่วนอื่น ๆ ของแผนที่ที่เราสนใจได้ เนื่องจากฝั่งพระประแดง เป็นพื้นที่ส่วนเขตฝั่งธนและบริเวณที่สนใจ ไม่มีการเชื่อมฝั่งกรุงเทพและฝั่งธนเข้าด้วยกัน จึงทำให้ข้อมูลสายรถเมล์ไม่ต่อเนื่องกัน นอกจากนี้แผนที่ใหญ่จะมีพื้นที่ด้านขวาสุดหายไปบางส่วน แต่ถือเป็นส่วนเล็กน้อยเนื่องจากสามารถเห็นโหนดได้ครบทั้งหมด ซึ่งเป็นข้อจำกัดของขนาดแผนที่ที่ใช้แต่เมื่อขยายขนาดเป็นรูปขนาดใหญ่แล้วจะสามารถเห็นแผนที่ได้ครบทั้งหมด นอกจากนี้ข้อมูลสายรถเมล์ที่ใช้ในการทำช่วงนี้จะใช้สายรถเมล์เพียงบางสายเท่านั้นที่ผ่านบนถนน โดยจะตัดรถเมล์สายที่วิ่งซ้ำกันออก เช่น สาย 206 กับ ปอ. 206 ซึ่งวิ่งในเส้นทางเดียวกัน หรือตัดสายที่วิ่งไปทางเดียวกันในอาณาเขตแผนที่นี้คือความจริงวิ่งคนละเส้นทางแต่เฉพาะในขอบเขตที่เราสนใจนี้ วิ่งบนเส้นทางเดียวกัน เช่น สาย 25 กับสาย 2 วิ่งบนถนนสุขุมวิทช่วงที่เราสนใจเหมือนกัน เราจึงเลือกมาเพียงสาย 2 สายเดียว

แต่เดิมเราได้จัดทำรถเมล์ไว้ทั้งหมด 15 สาย ได้แก่ สาย ปอ.1, 11, ปอ.14, ปอ.18, ปอ.19, ปอ.20, 22, 25, 40, 60, 92, 133, 137, 145 และสาย 206 แต่เมื่อมีการขยายพื้นที่ที่สนใจเพิ่มขึ้น เราจึงเพิ่มสายรถเมล์ในขอบเขตที่เราสนใจศึกษาอยู่ด้วยเพิ่มเป็น 35 สายได้แก่ สาย ปอ.1, 2, 4, ปอพ.6, 11, ปอ.12 ,ปอ.14, ปอ.15,ปอ.18, ปอ.19, ปอ.20, 22, 23 , 38, 40, 45, 46, 48, 71, 72, ปอ.92, 93, 98, 109, 113, ปอ.126, 133, 136, 137, 143, 145 และสาย 206 ซึ่งข้อมูลที่ใช้ทำจะมีความสะดวกขึ้นมากกว่าเดิม โดยหลักการเลือกสายที่สนใจจะเลือกสายที่ไม่มีเส้นทางเดินซ้ำกับสายอื่น มีเส้นทางเฉพาะในช่วงนี้ ตามที่ได้กล่าวมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

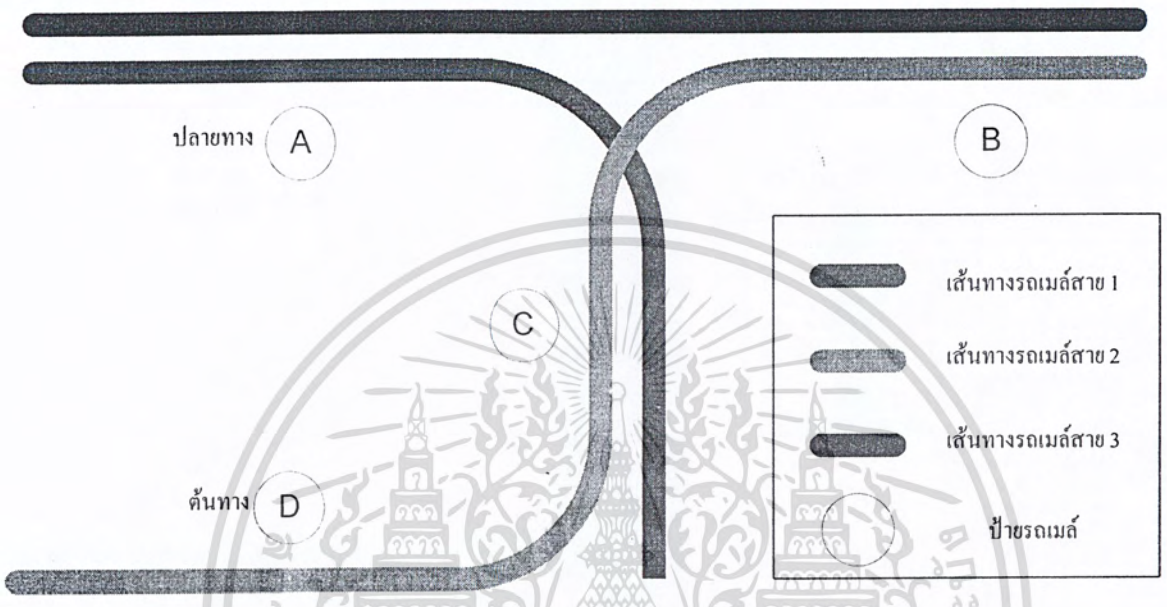
โดยรายละเอียดของรถเมล์และโหนดแต่ละโหนดเราจะได้กล่าวถึงในภาคผนวก โดยในภาคผนวกส่วนแรก จะได้กล่าวถึง รายละเอียดของสายรถเมล์ที่ใช้ในการทำงานทั้งหมดซึ่งมีรวม 35 สาย และในส่วนที่สองจะได้กล่าวถึงรายละเอียดของโหนดแต่ละส่วนว่ามีรายละเอียดเป็นอย่างไร แล้วโหนดดังกล่าวแทนถนนช่วงใดและแยกใด

เมื่อเราได้หลักการทำทั้งหมดแล้ว ต่อไปจะเป็นขั้นตอนการทำงานของโปรแกรมและผลลัพธ์ที่ได้ ในบทที่จะถึงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 8
# ตัวอย่างการทำงาน

## 8.1 ตัวอย่างการทำงาน




*รูปที่ 8-1 แผนที่แสดงเส้นทางรถเมล์และป้ายรถเมล์*

จากรูปที่ 8-1 จะได้ตาราง บัสไลน์ และบัสสต๊อป ดังนี้

| สายรถเมล์ | ผ่านป้ายรถเมล์ |
|---|---|
| 1 ไป | A > C |
| 1 กลับ | C > A |
| 2 ไป | D > C > B |
| 2 กลับ | B > C > D |
| 3 ไป | A > B |
| 3 กลับ | B > A |

| ป้ายรถเมล์ | สายรถเมล์ |
|---|---|
| A | 1 ไป, 1 กลับ, 3 ไป, 3 กลับ |
| B | 2 ไป, 2 กลับ, 3 ไป, 3 กลับ |
| C | 1 ไป, 1 กลับ, 2 ไป, 2 กลับ |
| D | 2 ไป, 2 กลับ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

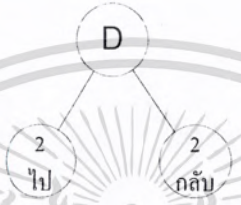
เนื่องจากเส้นทางเดินรถเมล์ในกรุงเทพ ฯ บางสาย ขาไปกับขากลับวิ่งไม่เหมือนกันจึงต้องทำให้มีสายรถเมล์ไปและกลับ

กำหนดให้ต้นทาง (start) เป็น : D

ปลายทาง (goal) เป็น : A

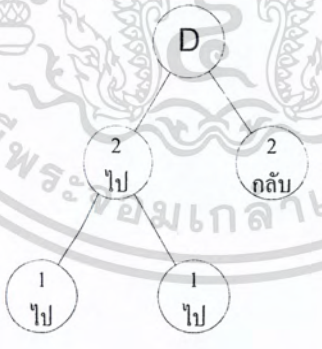
**8.1.1 การค้นหาเส้นทางโดยใช้อัลกอริทึม เบรดท์-เฟิรสท์ เซิรสช์**

ขั้นตอนที่ 1




- กำหนดต้นทาง (start) ป้ายรถเมล์ D เป็นรูทโหนดของทรี
- ดูว่าป้ายรถเมล์ D มีรถเมล์สายใดผ่านจากตารางบัสสต๊อป มีรถเมล์สาย 2 ไปและ 2 กลับผ่าน
- สร้างโหนดลูกของ D เป็นโหนดที่มีชื่อ 2 ไป และ 2 กลับ

ขั้นตอนที่ 2

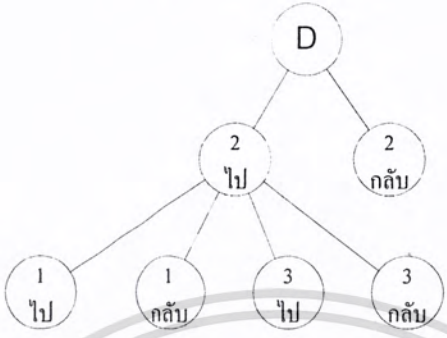



- ดูที่โหนดป้ายรถเมล์ D ว่าเป็นปลายทาง (goal) หรือไม่ ไม่ใช่ปลายทาง
- ดูที่โหนด 2 ไป แล้วมาดูในตารางบัสพาธว่ารถเมล์สาย 2 ไป นั้นวิ่งผ่านไปทางไหน
- จากตารางก็จะทราบว่าสาย 2 ไป นี้จะวิ่งจาก D ไป C ก็จะดูว่าที่ป้ายรถเมล์ C นั้นเป็นปลายหรือเปล่า จะได้ว่า C ไม่ใช่ปลายทาง
- ดูว่าในป้ายรถเมล์ C มีรถเมล์สายใดบ้างที่ยังไม่ได้สร้างเป็นโหนดในทรี
- จากตารางบัสสต๊อปก็จะทราบว่า รถเมล์สาย 1 ไป และ 1 กลับ ยังไม่ได้สร้างเป็นโหนดในทรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สร้างโหนด 1 ไป และ 1 กลับ เป็นโหนดลูกของ 2 ไป โดยในโหนด 1 ไป และ 1 กลับ นั้นจะต้องเก็บค่าของป้ายรถเมล์ C ด้วยเพื่อจะได้ทราบว่ามีการต่อรถที่ป้ายรถเมล์ใด

ขั้นตอนที่ 3




- จากนั้นรถเมล์สาย 2 ไป ก็จะไปจากป้ายรถเมล์ C ไป B ก็จะดูว่าที่ป้ายรถเมล์ B เป็นปลายทางหรือไม่ ไม่เป็นปลายทาง
- ดูว่ามีรถเมล์สายใดในป้ายรถเมล์ B ที่ยังไม่ได้สร้างเป็นโหนด ก็จะมีสาย 3 ไป และ 3 กลับ
- สร้างโหนด 3 ไป และ 3 กลับ เป็นโหนดลูกของ 2 ไป โดยเก็บค่าป้ายรถเมล์ B ในโหนดด้วย

ขั้นตอนที่ 4

- ดูในตารางบัสพาธทราบว่าสาย 2 ไป สุดเส้นทางแล้ว
- ไปดูที่โหนด 2 กลับ (ตามหลัก เบรดท์-เฟิรสท์ เซิรสช์) แล้วดูในตารางบัสพาธอีกก็จะทราบว่าสาย 2 กลับ นั้นสุดเส้นทางแล้ว
- ตัวค้นหาเส้นทางก็จะไปที่โหนด 1 ไป แล้วไปดูในตารางบัสพาธว่าสาย 1 ไป นั้นสุดเส้นทางแล้ว
- ไปดูที่โหนด 1 กลับ ตัวค้นหาเส้นทางดูในตารางบัสพาธ พบว่าสาย 1 ไป วิ่งจากป้ายรถเมล์ C ไป A ก็จะดูว่าเป็นปลายทางหรือเปล่า พบว่าเป็นปลายทางจึงได้เส้นทาง

  สาย 2 ไป : ผ่านป้าย D > C

  สาย 1 กลับ : ผ่านป้าย C > A เป็นคำตอบอันแรก
- ดูว่าในโหนด 3 ไป พบว่าสุดเส้นทางแล้ว
- ไปดูที่โหนด 3 กลับ ดูในตารางบัสพาธก็จะทราบว่าสาย 3 กลับ วิ่งจากป้าย B ไป A ก็จะดูว่าป้าย A เป็นปลายทางหรือเปล่า ได้ว่าเป็นปลายทางจึงได้เส้นทาง

  สาย 2 ไป : ผ่านป้าย D > C > B

  สาย 3 กลับ : ผ่านป้าย B > A เป็นคำตอบที่ 2
- ส่งคำตอบทั้ง 2 ไปยังส่วนติดต่อกับผู้ใช้เพื่อแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**8.1.2 การค้นหาเส้นทางโดยใช้อัลกอริทึมอิเทอเรทีฟ ดีพเพ็นดิ่ง เซิรสช์**

ขั้นตอนที่ 1

Depth = 0

- กำหนดต้นทาง (start) ป้ายรถเมล์ D เป็นรูทโหนดของทรี
- ดูว่าในระดับเดพท์ (depth) = 0 มีโหนดอื่นอีกหรือไม่ (กำหนด depth limited แรก = 0) ไม่มี เริ่มทำใหม่โดยให้เดพท์เพิ่มขึ้น 1 (depth = depth+1)

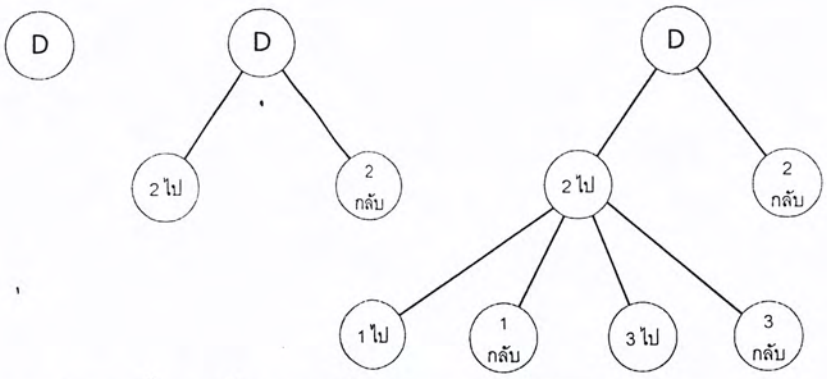
ขั้นตอนที่ 2

Depth = 1

- ทำตามขั้นตอนที่ 1 สร้างรูทโหนดของทรี
- ดูว่าเดพท์ที่อยู่ = เดพท์ลิมิตแล้วหรือยัง ไม่
- ดูว่าป้าย D มีรถเมล์สายไหนผ่าน มีสาย 2 ไป และสาย 2 กลับ
- สร้างโหนด 2 ไป และ 2 กลับ เป็นโหนดลูกของรูทโหนด D
- ไปยังโหนด 2 ไป ในระดับชั้นถัดไป ดูว่า 2 ไป ผ่านป้ายอะไรบ้าง ผ่านป้าย C และ ป้าย B
- ป้าย C เป็นปลายทางหรือไม่ ไม่ ป้าย B เป็นปลายทางหรือไม่ ไม่
- ดูว่าเดพท์ที่อยู่ = เดพท์ลิมิตหรือไม่ ,ถึงแล้ว ไม่มีการขยายโหนดต่อ
- กลับไปยังโหนดแม่ คือ รูทโหนดดูว่ามีโหนดลูกใดที่ยังไม่ทำการค้นหา มีโหนด 2 กลับ
- ไปยังโหนด 2 กลับ ดูว่า 2 กลับ ผ่านป้ายอะไรบ้าง ไม่ผ่าน(สุดสายแล้ว)
- เดพท์ = 1 มีโหนดอื่นอีหรือไม่ ไม่มี
- เริ่มใหม่เพิ่มค่าเดพท์อีกหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 3




Depth = 2

- ทำตามขั้นตอนที่ 2 จนกระทั่งถึงโหนดสาย 2 ไป
- ดูว่าเดพท์ที่อยู่ = เดพท์ลิมิตหรือไม่ ,ไม่ ดังนั้นจึงมีการขยายโหนดต่อ
- สาย 2 ไป ผ่านป้าย C เป็นปลายทางหรือไม่ ไม่ใช่
- ที่ป้าย C มีรถเมล์ที่ยังไม่เคยขึ้นหรือไม่ มีคือสาย 1 ไป, สาย 1 กลับ
- สร้างโหนด 1 ไป, 1 กลับ เป็นโหนดลูกของสาย 2 ไป เก็บป้าย C ไว้เป็นจุดที่ขึ้นรถ
- สาย 2 ไปยังป้าย B เป็นปลายทางหรือไม่ ไม่ใช่
- ที่ป้าย B มีรถเมล์ที่ยังไม่เคยขึ้นหรือไม่ มีคือสาย 3 ไป,สาย 3 กลับ
- สร้างโหนด 3 ไป, 3 กลับ เป็นโหนดลูกของสาย 2 ไป เก็บป้าย B ไว้เป็นจุดขึ้นรถ
- ไปยังโหนด 1 ไป ดูว่าผ่านปลายทางหรือไม่ ไม่ผ่าน
- ดูว่าเดพท์ที่อยู่ = เดพท์ลิมิตหอไม่, ถึงแล้ว ไม่มีการขยายโหนดต่อ
- กลับไปยังโหนด 2 ไปดูว่ามีโหนดลูกใดที่ยังไม่ค้นหา มี
- ไปยังโหนด 1 กลับ ดูว่าผ่านปลายทางหรือไม่ ,ผ่าน ส่งค่าเส้นทางที่ 1 คือ

  สาย 2 ไป : ผ่านป้าย D > C

  สาย 1 กลับ : ผ่านป้าย C > A เป็นคำตอบอันแรก
- กลับไปโหนด 2 ไปดูว่ามีโหนดลูกใดที่ยังไม่ค้นหา มี
- ไปยังโหนด 3 ไป ดูว่าผ่านปลายทางหรือไม่ ไม่ผ่าน
- กลับไปโหนด 2 ไปดูว่ามีโหนดลูกใดที่ยังไม่ค้นหา มี
- ไปยังโหนด 3 กลับ ดูว่าผ่านปลายทางหรือไม่ ,ผ่าน ส่งค่าเส้นทางที่ 2 คือ

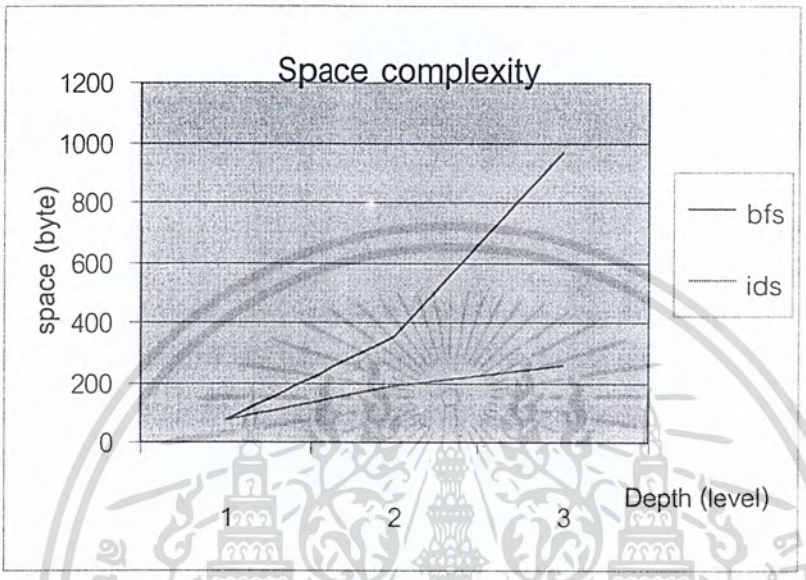
  สาย 2 ไป : ผ่านป้าย D > C > B

  สาย 3 กลับ : ผ่านป้าย B > A เป็นคำตอบที่ 2
- กลับไปโหนด 2 ไปดูว่ามีโหนดลูกใดที่ยังไม่ค้นหา ไม่มี
- กลับไปโหนด D ดูว่ามีโหนดลูกใดที่ยังไม่ได้ค้นหา มี
- ไปยังโหนด 2 กลับ ดูว่าผ่านป้ายใด ไม่ผ่าน(สุดสาย)
- กลับไปโหนด D ดูว่ามีโหนดลูกใดที่ยังไม่ได้ค้นหา ไม่มี
- ส่งคำตอบทั้งหมดไปให้ส่วนติดต่อกับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**8.2 ผลการทดสอบวิธีการค้นหาทั้ง 2 วิธี**

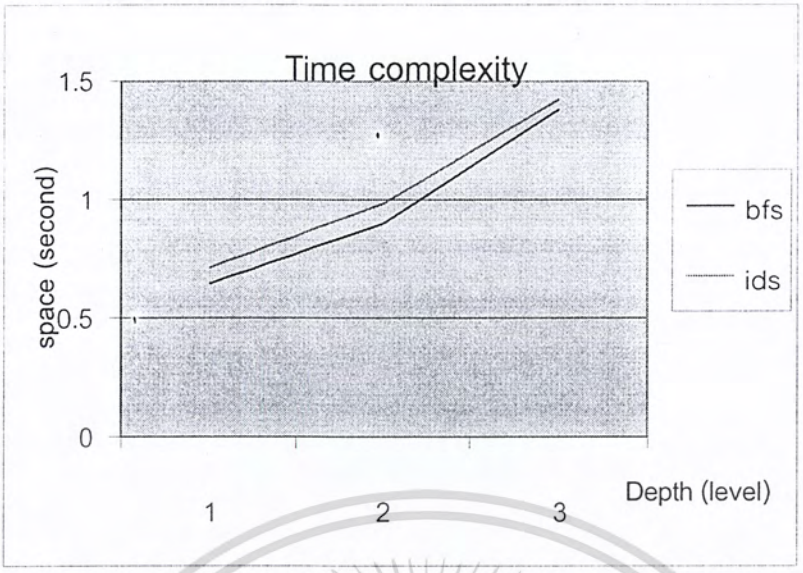
จุดประสงค์หนึ่งในการทำโครงงานนี้ก็คือ การเปรียบเทียบประสิทธิภาพอัลกอริทึมที่ใช้ในการค้นหา จากที่ทราบในบทที่ 5 แล้วว่ามีตัววัดประสิทธิภาพการค้นหาอยู่ 4 ตัว แต่ที่เราจะมาดูกันในที่นี้ คือ ความซับซ้อนของเนื้อที่หน่วยความจำ (Space complexity) และ ความซับซ้อนของเวลา (Time complexity)




*รูปที่ 8-2 กราฟแสดงการเปรียบเทียบหน่วยความจำที่ใช้ระหว่างอัลกอริทึมเบรดท์-เฟิรสท์ เซิรสช์ กับอัลกอริทึมอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์*

จากรูปที่ 8-2 จะเห็นได้ว่ากราฟของเบรดท์-เฟิรสท์ เซิรสช์ จะพุ่งขึ้นในลักษณะที่เป็นเอ็กโปเน็นเชี่ยล ถ้าเดพท์เพิ่มขึ้น แต่กราฟของอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ จะเป็นลักษณะเชิงเส้น โดยที่เดพท์ = 1 จะใช้หน่วยความจำเท่ากับเบรดท์-เฟิรสท์ เซิรสช์ แต่เมื่อมีเดพท์มากขึ้น อัลกอริทึมอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ จะใช้หน่วยความจำน้อยกว่าแบบเบรดท์-เฟิรสท์ มาก ซึ่งตรงกับทฤษฎี เพราะเบรดท์-เฟิรสท์ เซิรสช์ ใช้หน่วยความจำเป็นจำนวน $O(b^d)$ กราฟจึงขึ้นในแบบเอ็กโปเน็นเชี่ยล ส่วนอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ ใช้หน่วยความจำเป็นจำนวน O(bd) กราฟจึงขึ้นในลักษณะเชิงเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




*รูปที่ 8-3 กราฟแสดงการใช้เวลาที่ใช้ระหว่างอัลกอริทึมเบรดท์-เฟิรสท์ เซิรสช์ กับอัลกอริทึมอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์*

จากรูปที่ 8-3 จะเห็นได้ว่ากราฟทั้งแบบเบรดท์-เฟิรสท์ เซิรสช์ และอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ จะขึ้นไปเป็นลักษณะเอ็กโปเน็นเชี่ยล เพราะจากทฤษฎีในบทที่ 5 จะทราบได้ว่า การใช้เวลาเป็น $O(b^d)$ เหมือนกัน แต่อัลกอริทึมแบบอิเทอเรทีฟ ดิพเพ็นนิ่ง เซิรสช์ จะใช้เวลามากกว่าแบบเบรดท์-เฟิรสท์ เซิรสช์เล็กน้อยเพราะจะมีการค้นหาที่ซ้ำกันในความลึกระดับแรก แต่จะมากกว่ากันไม่มากเพราะการค้นหาในความลึกระดับแรกจะมีโหนดไม่มากเมื่อเทียบกับในโหนดระดับลึก ๆ โดยทางทฤษฎีแล้วจะใช้เวลาไม่เกิน 2 เท่าของเบรดท์-เฟิรสท์ เซิรสช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 9
# วิจารณ์และสรุป

จากการที่เราได้ศึกษาข้อมูลต่าง ๆ ของสายรถเมล์ ทำให้เราได้พบปัญหาจำนวนมากในการทำงาน เนื่องจากการจัดสายรถเมล์ในปัจจุบัน ยังไม่มีระบบที่ดีในการรองรับกับสายรถเมล์ที่มีอยู่เป็นจำนวนมากและมีความกระจัดกระจายของสายรถเมล์มาก รวมถึงมีความซ้ำซ้อนที่เกิดขึ้น และอีกปัญหาหนึ่งคือด้วยรูปลักษณะของถนน ที่จะมีความยุ่งยากในตัวเอง กล่าวคือจะมีลักษณะที่เป็นวันเวย์ ทำให้ระบบสายรถเมล์มีเส้นทางที่ไม่เหมือนกัน ทั้งสายไปและสายกลับ มีความแตกต่าง และเนื่องจากเหตุผลดังกล่าวมาข้างต้นนี้ จึงทำให้การใช้อัลกอริทึมในการเซิรสช์เกิดปัญหาขึ้นบ้าง โดยเราจะพยายามแก้ไขปัญหาโดยการจะตัดขอบเขตของปัญหาให้เหลือเพียงพอที่จะรองรับได้ แต่ยังอยู่ในขอบเขตที่พอสมควร ไม่ตัดออกจนเกินไป ซึ่งจะสามารถแก้ไขปัญหาเล็ก ๆ น้อย ๆ เหล่านี้ได้มากขึ้น

ในส่วนของภาษาที่ใช้ในการเขียนโปรแกรมนี้ เราได้ศึกษา แล้วพบว่าจะเป็นการสะดวกหากมีการดำเนินการโดยใช้ภาษา จาวา ทั้งในส่วนของอัลกอริทึมการค้นหา และส่วนของการรับอินพุต และแสดงผล แต่เนื่องจากภาษาจาวาเป็นภาษาที่ใหม่สำหรับผู้จัดทำ จึงอาจจะมีข้อผิดพลาดในการเขียนโปรแกรมโดยใช้ภาษา จาวาบางส่วน

โดยสรุปจากการทำงานที่ผ่านมา เราได้พบปัญหาในหลาย ๆ ส่วน ทั้งในเรื่องของการเขียนโปรแกรมจาวาให้อยู่ในรูปแบบของอัลกอริทึมที่ดี ปัญหาเรื่องขอบเขตที่สนใจ ปัญหาเรื่องข้อมูลที่จะนำมาใช้ ปัญหาต่าง ๆ เหล่านี้ได้ถูกพบระหว่างการทำโครงงานชิ้นนี้ แต่ก็ได้รับการแก้ไขไปพอสมควร จนทำให้ได้ความรู้ในเรื่องต่าง ๆ กลับมามาก จึงถือว่าการทำโครงงานชิ้นนี้ได้สอนทางอ้อมในเรื่องต่าง ๆ หลายหลาย เรื่องให้กับผู้ทำ

ด้านการประยุกต์ใช้ ต่อไปในอนาคตนั้น สามารถทำได้โดยการพัฒนาฐานข้อมูลให้มีจำนวนโหนดที่สนใจ จำนวนขอบเขตของแผนที่ที่สนใจให้มากยิ่งขึ้นกว่าในปัจจุบัน และยังสามารถนำไปประยุกต์ใช้เข้ากับแผนที่แบบอื่น ได้ต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก
# รายละเอียดรถเมล์แต่ละสายที่ใช้ในโครงงาน

รถเมล์หมายเลข 1 คือหมายเลขแทนรถเมล์สาย ปอ. 1 ถ้าดูจากขอบเขตในแผนที่จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่ง จากถนนสุขาภิบาล 2 คือโหนด 2 ไปเรื่อย ๆ ตามถนนรามคำแหง ไปจนสิ้นสุดที่ถนนสุขุมวิท ช่วงนานา ตามจุดสิ้นสุดแผนที่ คือโหนด 48

รถเมล์หมายเลข 2 คือหมายเลขแทนรถเมล์สาย 2 ถ้าดูจากขอบเขตในแผนที่จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่ง จาก ถนนสุขุมวิท ช่วงนานา คือโหนด 48 ไปเรื่อย ๆ ตามถนนสุขุมวิท ไปจนสิ้นสุดที่ถนนสุขุมวิท ผ่านบางนา จนถึงสำโรง ตามจุดสิ้นสุดแผนที่ คือโหนด 62

รถเมล์หมายเลข 4 คือหมายเลขแทนรถเมล์สาย 4 ถ้าดูจากขอบเขตในแผนที่จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่ง จากถนนพระราม 4 ช่วงสวนลุม คือโหนด 51 ไปยังแยกคลองเตย ไปจนสิ้นสุดที่ถนนสุนทรโกษา เพื่อเตรียมตัวเข้าอู่ท่าเรือคลองเตย ตามจุดสิ้นสุดแผนที่ คือโหนด 54 ซึ่งเป็นจุดบนถนนสุนทรโกษา

รถเมล์หมายเลข 6 คือหมายเลขแทนรถเมล์สาย ป.อ.พ. 6 ถ้าดูจากขอบเขตในแผนที่จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่ง จากถนนสุขุมวิท ช่วงนานา คือโหนด 2 ไปเรื่อย ๆ ตามถนนสุขุมวิท ไปจนสิ้นสุดที่ถนนศรีนครินทร์ ช่วงเทพารักษ์ ตามจุดสิ้นสุดแผนที่ คือโหนด 63

รถเมล์หมายเลข 11 คือหมายเลขแทนรถเมล์สาย 11 ถ้าดูจากขอบเขตในแผนที่จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนเพชรบุรี ช่วงมักกะสัน คือโหนด 45 ไปเรื่อย ๆ ตามถนนเพชรบุรี ไปจนสิ้นสุดที่ถนนอ่อนนุช ลาดกระบัง ช่วงประเวศ ตามจุดสิ้นสุดแผนที่ คือโหนด 23

รถเมล์หมายเลข 12 คือหมายเลขแทนรถเมล์สาย ปอ.12 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนเพชรบุรี ช่วงมักกะสัน คือโหนด 45 ไปเรื่อย ๆ ตามถนนเพชรบุรี เลี้ยวไปทางถนนรามคำแหง ไปจนถึงสี่แยกลำสาลี จากนั้นจะไปจนสิ้นสุดที่ถนนสุขาภิบาล 2 ตามจุดสิ้นสุดแผนที่ คือโหนด 2

รถเมล์หมายเลข 14 คือหมายเลขแทนรถเมล์สาย ปอ.14 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนดินแดง ผ่านมาทาง อ.ส.ม.ท. คือโหนด 42 ไปเรื่อย ๆ ตามถนนพระราม 9 ไปรามคำแหง จากนั้นจะไปจนสิ้นสุดที่ถนนสุขาภิบาล 3 ตามจุดสิ้นสุดแผนที่ คือโหนด 3

รถเมล์หมายเลข 15 คือหมายเลขแทนรถเมล์สาย ปอ.15 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนดินแดง คือโหนด 42 มายังแยก อ.ส.ม.ท. เลี้ยวซ้ายไปจนสิ้นสุดที่ถนนรัชดา ช่วงฟอร์จูนทาวเวอร์ ตามจุดสิ้นสุดแผนที่ คือโหนด 41

รถเมล์หมายเลข 18 คือหมายเลขแทนรถเมล์สาย ปอ.18 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนรัชดาภิเษก ช่วงฟอร์จูน คือโหนด 41 ไปเรื่อย ๆ ตามถนนดินแดง จนถึงสิ้นสุดที่จุดถนนอ่อนนุช บริเวณ ประเวศ ตามจุดสิ้นสุดแผนที่ คือโหนด 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถเมล์หมายเลข 19 คือหมายเลขแทนรถเมล์สาย ปอ. 19 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขาภิบาล 2 คือโหนด 2 ไปเรื่อย ๆ ตามสี่แยกลำสาลี ไปสุดทางที่ถนนลาดพร้าว ตามจุดสิ้นสุดแผนที่ คือโหนด 1

รถเมล์หมายเลข 20 คือหมายเลขแทนรถเมล์สาย ปอ. 20 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขาภิบาล 3 คือโหนด 3 ไปเรื่อย ๆ ตามถนนศรีนครินทร์ ไปตามถนนพระราม 4 ผ่านแยกคลองเตย ตามจุดสิ้นสุดแผนที่ คือโหนด 53

รถเมล์หมายเลข 22 คือหมายเลขแทนรถเมล์สาย 22 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขุมวิท ช่วงนานา คือโหนด 2 ไปเรื่อย ๆ ตามถนนสุขุมวิท ไปจนสิ้นสุดที่ถนนสุขุมวิท ผ่านบางนา จนถึงสำโรง ตามจุดสิ้นสุดแผนที่ คือโหนด

รถเมล์หมายเลข 23 คือหมายเลขแทนรถเมล์สาย 23 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนเพชรบุรี 45 ไปเรื่อย ๆ ตามถนนเพชรบุรี ไปจนสิ้นสุดที่ถนนสุขุมวิท ผ่านบางนา จนถึงสำโรง ตามจุดสิ้นสุดแผนที่ คือโหนด 62

รถเมล์หมายเลข 38 คือหมายเลขแทนรถเมล์สาย 38 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนเพชรบุรี ช่วงมักกะสัน คือโหนด 2 ไปเรื่อย ๆ ตามถนนสุขุมวิท ไปจนสิ้นสุดที่ถนนบางนา ผ่าน กม. 4 จนถึง ม. ราม 2 ตามจุดสิ้นสุดแผนที่ คือโหนด 63

รถเมล์หมายเลข 40 คือหมายเลขแทนรถเมล์สาย 40 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขุมวิท ช่วงนานา คือโหนด 48 ไปเรื่อย ๆ ตามถนนสุขุมวิท ไปจนถึงถนนศรีนครินทร์ ผ่านแยกอุดมสุข จนถึงถนนอุดมสุข ตามจุดสิ้นสุดแผนที่ คือโหนด 56

รถเมล์หมายเลข 45 คือหมายเลขแทนรถเมล์สาย 45 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนพระราม 4 ช่วงสวนลุม คือโหนด 51 ไปเรื่อย ๆ ตามถนนพระราม 4 ไปจนถึงถนนสุขุมวิท ผ่านแยกบางนา จนถึงสิ้นสุดที่ ถนนสุขุมวิท ช่วงสำโรง ตามจุดสิ้นสุดแผนที่ คือโหนด 62

รถเมล์หมายเลข 46 คือหมายเลขแทนรถเมล์สาย 46 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนพระราม 4 ช่วงสวนลุม คือโหนด 51 ไปเรื่อย ๆ ตามถนนพระราม 4 ไปจนถึงถนนสุขุมวิท ผ่านแยกบางนา เลี้ยวไปตามถนน บางนา-ตราด จนถึงสิ้นสุดที่ ถนนบางนา ช่วง มหาวิทยาลัย รามคำแหง 2 ตามจุดสิ้นสุดแผนที่ คือโหนด 61

รถเมล์หมายเลข 48 คือหมายเลขแทนรถเมล์สาย 48 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขุมวิท ช่วงนานา คือโหนด 48 ไปเรื่อย ๆ ตามถนนสุขุมวิท ไปจนถึงถนนสุขุมวิท ช่วงที่ผ่านแยกบางนา เลี้ยวไปตามถนน บางนา-ตราด จนถึงสิ้นสุดที่ ถนนบางนา ช่วง มหาวิทยาลัย รามคำแหง 2 ตามจุดสิ้นสุดแผนที่ คือโหนด 61

รถเมล์หมายเลข 71 คือหมายเลขแทนรถเมล์สาย 71 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขาภิบาล 2 คือโหนด 2 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนรามคำแหง ไปจนถึงถนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุขุมวิท 71 ผ่านแยกพระโขนง-พระราม 4 เลี้ยวไปตามถนน พระราม 4 จนถึงสิ้นสุดที่ ถนนสุขุมวิท 42 หรือกล้วยน้ำไท ที่เป็นอู่รถเมล์ ตามจุดในแผนที่ คือโหนด 33

รถเมล์หมายเลข 72 คือหมายเลขแทนรถเมล์สาย 72 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนเพชรบุรี คือโหนด 45 ไปทางสี่แยกอโศกเพชรบุรี ไปเรื่อย ๆ ตามเพชรบุรี ไปจนถึงถนนสุขุมวิท 63 ผ่านแยกเอกมัยสุขุมวิท เลี้ยวไปตามถนน ถนนสุขุมวิท 42 หรือกล้วยน้ำไท จนไปสิ้นสุดที่ถนนอาจณรงค์ ตามจุดในแผนที่ คือโหนด 37

รถเมล์หมายเลข 92 คือหมายเลขแทนรถเมล์สาย ๋อ. 92 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนลาดพร้าว คือโหนด 1 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนรามคำแหง ไปจนถึงถนนพัฒนาการ ผ่านแยกพัฒนาการ เลี้ยวไปตามถนนอ่อนนุช จนไปสิ้นสุดที่ ถนนอ่อนนุช ช่วง ประเวศ ตามจุดในแผนที่ คือโหนด 23

รถเมล์หมายเลข 93 คือหมายเลขแทนรถเมล์สาย 93 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนเพชรบุรี คือโหนด 45 ไปทางสี่แยกอโศกเพชรบุรี ไปเรื่อย ๆ ตามเพชรบุรี และไปตามถนนรามคำแหง จนถึงสิ้นสุดที่ หมู่บ้านนักกีฬา ที่เป็นอู่รถเมล์ ตามจุดในแผนที่ คือโหนด 6

รถเมล์หมายเลข 98 คือหมายเลขแทนรถเมล์สาย 98 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนดินแดง คือโหนด 42 ไปทางสี่แยกอ.ส.ม.ท. ไปเรื่อย ๆ ตามถนนอโศก และไปตามถนนสุขุมวิทจนถึงสิ้นสุดที่ อู่กล้วยน้ำไท ที่เป็นอู่รถเมล์ ตามจุดในแผนที่ คือโหนด 33

รถเมล์หมายเลข 109 คือหมายเลขแทนรถเมล์สาย 109 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขาภิบาล 2 คือโหนด 2 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนรามคำแหง และไปตามถนน พระราม 4 จนถึงสิ้นสุดที่ พระราม4 ช่งสวนลุม ตามจุดในแผนที่ คือโหนด 51

รถเมล์หมายเลข 113 คือหมายเลขแทนรถเมล์สาย 113 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนสุขาภิบาล 3 คือโหนด 3 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนรามคำแหง และไปตามถนน เพชรบุรี จนถึงสิ้นสุดที่ เพชรบุรี ช่วง มักกะสัน ตามจุดในแผนที่ คือโหนด 45

รถเมล์หมายเลข 126 คือหมายเลขแทนรถเมล์สาย ปอ. 126 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนลาดพร้าว คือโหนด 1 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนรามคำแหง และไปตามถนน สุขุมวิท จนถึงสิ้นสุดที่ สุขุมวิท ช่วง สำโรง ตามจุดในแผนที่ คือโหนด 62

รถเมล์หมายเลข 133 คือหมายเลขแทนรถเมล์สาย 133 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนบางนา-ตราด คือโหนด 61 ไปทางสี่แยกบางนา ไปเรื่อย ๆ ตามถนนศรีนครินทร์ และไปตามถนน สุขุมวิท 71 และ พระราม 4จนถึงสิ้นสุดที่ สุขุมวิท 42 ซึ่งเป็นอู่ ตามจุดในแผนที่ คือโหนด 33

รถเมล์หมายเลข 136 คือหมายเลขแทนรถเมล์สาย 136 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนรัชดาภิเษก คือโหนด 41 ไปทางสี่แยก อ.ส.ม.ท. ไปเรื่อย ๆ ตามถนนรัชดาภิเษก จนถึงสิ้นสุดที่ ถนน สุนทรโกษา ตามจุดในแผนที่ คือโหนด 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รถเมล์หมายเลข 137 คือหมายเลขแทนรถเมล์สาย 137 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนลาดพร้าว คือโหนด 1 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนรามคำแหง และไปตามถนนดินแดง จนถึงสิ้นสุดที่ ถนนรัชดาภิเษก ช่วง ฟอร์จูน ตามจุดในแผนที่ คือโหนด 41

รถเมล์หมายเลข 143 คือหมายเลขแทนรถเมล์สาย 143 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนลาดพร้าว คือโหนด 1 ไปทางสี่แยกลำสาลี และไปจนถึงสิ้นสุดที่ สุขาภิบาล 3 ตามจุดในแผนที่ คือโหนด 3

รถเมล์หมายเลข 145 คือหมายเลขแทนรถเมล์สาย 145 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนลาดพร้าว คือโหนด 1 ไปทางสี่แยกลำสาลี ไปเรื่อย ๆ ตามถนนศรีนครินทร์ จนถึงสิ้นสุดที่ ศรีนครินทร์ ช่วง เทพารักษ์ ตามจุดในแผนที่ คือโหนด 63

รถเมล์หมายเลข 149 คือหมายเลขแทนรถเมล์สาย 149 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนพระราม 4 คือโหนด 51 ไปทางสี่แยกคลองเตย ไปเรื่อย ๆ ตามถนนพระราม 4 จนถึงสิ้นสุดที่ ถนน อาจณรงค์ ตามจุดในแผนที่ คือโหนด 37

รถเมล์หมายเลข 205 คือหมายเลขแทนรถเมล์สาย 205 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนพระราม 3 คือโหนด 53 ไปทางสี่แยกคลองเตย จนถึงสิ้นสุดที่ ถนน สุนทรโกษา ตามจุดในแผนที่ คือโหนด 54

รถเมล์หมายเลข 206 คือหมายเลขแทนรถเมล์สาย 206 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนรัชดาภิเษก คือโหนด 41 ไปทางสี่แยก อ.ส.ม.ท. ไปเรื่อย ๆ ตามถนนรัชดาภิเษก และไปตามถนน เพชรบุรี ศรีนครินทร์ จนถึงสิ้นสุดที่ อุดมสุข ตามจุดในแผนที่ คือโหนด 56

รถเมล์หมายเลข 207 คือหมายเลขแทนรถเมล์สาย 207 ถ้าดูจากขอบเขตในแผนที่ จะเห็นว่ารถเมล์สายนี้จะเริ่มวิ่งจากถนนบางนา-ตราด คือโหนด 61 ไปทางสี่แยกบางนา ไปเรื่อย ๆ ตามถนนศรีนครินทร์ และไปตามถนน รามคำแหง จนถึงสิ้นสุดที่ อู่รามคำแหง ที่เป็นอู่รถเมล์ ตามจุดในแผนที่ คือโหนด 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข
# รายละเอียดโหนดที่แทนแต่ละป้ายในโครงงาน

1. ลาดพร้าว ถ้าดูจากในแผนที่ จะเห็นว่าเริ่มจากสุดแผนที่ จนถึงสี่แยกลำสาลี แต่ในความเป็นจริงแล้วจะเริ่มจากสี่แยกรัชดาลาดพร้าว ถึงสี่แยกลำสาลี โดยจะมีอู่รถอยู่รหว่างทาง 2 อู่คือ 22 และ สาย 143 ซึ่งจไม่ได้สนใจในขอบเขตนี้
2. สุขาภิบาล 2 ถ้าดูจากในแผนที่จะเห็นส่วนของถนนเส้นนี้มีส่วนเล็ก ๆ ความจริงเริ่มจากนิด้า จนถึงสี่แยกลำสาลี
3. สุขาภิบาล 3 จะเริ่มจากมีนบุรี มาจนถึง สี่แยกลำสาลี แต่ในแผนที่จะให้เห็นแค่เพียงบางส่วนของถนนสุขาภิบาล 3 เท่านั้น
4. สี่แยกลำสาลี เป็นสี่แยก แต่ในแผนที่ที่คิดนั้น ให้สี่แยกลำสาลี เป็น 5 แยก โดยรวมแยกรอบ ๆ ใกล้ ๆ มารวมกันให้เป็น 5 แยก ความจริงยังต้องรวม ถ.สุขาภิบาล 1 อีก 1 สาย เป็น 6 แยก แต่เนื่องจากช่วงถนนของสุขาภิบาล 1 ที่ปรากฏในแผนที่มีน้อยมากจึงทำการตัดออก
5. รามคำแหง (ม.ราม) เป็นรามคำแหงช่วง มหาวิทยาลัยรามคำแหง โดยเริ่มจากสี่แยกพระราม9 รามคำแหงถึง สี่แยกลำสาลี เป็นถนนที่มีรถเมลีผ่านจำนวนหลายสาย และมีอู่รถอยู่ 1 สายที่สนใจ คือ สาย 207 อยู่บริเวณ มหาวิทยาลัย รามคำแหง
6. ศรีนครินทร์(ม.นักกีฬา) เป็นถนนศรีนครินทร์ช่วงต้นเริ่มจากสี่แยกลำสาลีไปจนถึงแยกพระราม 9 ศรีนครินทร์ ในระหว่างทางนั้นจะมีอู่รถสาย 93 อยู่บริเวณหมู่บ้านนักกีฬา
7. พระราม 9 เริ่มจากสี่แยก อ.ส.ม.ท. ไปสิ้นสุดที่ สี่แยกพระราม 9 รามคำแหง ระหว่างทางจะมีถนนตัดผ่านอยู่แต่ เราไม่ได้นำมาสนใจ
8. สี่แยกพระราม9-รามคำแหง เกิดจากการตัดกันของถนนรามคำแหงกับถนนพระราม 9 เดิมและพระราม 9 ตัดใหม่
9. พระราม 9 ตัดใหม่ เป็นถนนที่ตัดใหม่ เริ่มจาก สี่แยกพระราม 9 รามคำแหง ไปสิ้นสุดที่ สี่แยกพระราม 9 ศรีนครินทร์ ปัจจุบันมีรถเมล์เพียงสายเดียวที่วิ่งผ่านคือสาย 40
10. ศรีนครินทร์(สถานีรถไฟหัวหมาก) เป็นถนนช่วงสั้น ๆ ที่อยู่ระหว่างสี่แยกพระราม 9-ศรีนครินทร์ กับสี่แยกพัฒนาการ
11. รามคำแหง(รพ.แพทย์ปัญญา) ถนนรามคำแหงช่วงต้น เริ่มจากสี่แยกคลองตัน ไปยังสี่แยกรามคำแหง-พระราม9
12. เพชรบุรีตัดใหม่(คลองตัน)ถนนเพชรบุรีตัดใหม่ช่วงคลองตัน คือเริ่มจาก แยกเอกมัยเพชรบุรี ไปจนถึงสี่แยกคลองตัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. สี่แยกคลองตัน เกิดจากสี่ถนนมาบรรจบกัน คือตามแนวนอนคือถนนเพชรบุรีตัดใหม่ กับถนนพัฒนาการ และแนวตั้งคือถนนรามคำแหง และถนนสุขุมวิท 71
14. สุขุมวิท 71 หรือถนนพระโขนง-คลองตัน เริ่มจาก แยกพระโขนงพระราม 4 ไปสิ้นสุดที่สี่แยกคลองตัน
15. พัฒนาการ(ม.เกษมบัณฑิต) เป็นพัฒนาการช่วงต้น เริ่มจากสี่แยกคลองตัน ไปยังสี่แยกพัฒนาการ โดยมีอู่รถสาย 92 อยู่ แต่เราไม่ได้สนใจสายนี้เนื่องจากเราสนใจรถ ปอ. 92 ที่มีระยะทางวิ่งไกลกว่า
16. สี่แยกพัฒนาการ เป็นสี่แยกที่เกิดจากการตัดกันของถนนพัฒนาการและถนนศรีนครินทร์ เรียกอีกชื่อว่าสี่แยกศรีพัฒน์
17. พัฒนาการ(ม.ผาสุก) เป็นพัฒนาการช่วงปลายถนน จะมีอู่สาย 11 คันเล็กอยู่ปลายถนน แต่เราไม่ได้นำมาสนใจคิดด้วย เริ่มจากสี่แยกพัฒนาการ ถึง สามแยกผาสุก
18. ศรีนครินทร์(ตลาดเอี่ยมสมบัติ) เป็นช่วงถนนศรีนครินทร์ ที่เริ่มจากสี่แยกพัฒนาการ มาสิ้นสุดที่สี่แยกอ่อนนุช
19. สุขุมวิท (พระโขนง) เป็นถนนสุขุมวิทช่วงสั้น ๆ อยู่ระหว่าง แยกพระโขนง และสามแยกอ่อนนุช มีอู่สายรถเมล์ 132 และ 1013 อยู่ แต่ไม่ได้สนใจนำมาคิดเช่นกัน
20. สุขุมวิท (บางจาก) เป็นถนนสุขุมวิทช่วงตั้งแต่สามแยกอ่อนนุช จนถึง แยกบางนา ระหว่างทางจะมีทางด่วนสุขุมวิท 62 แต่เราไม่ได้สนใจในเรื่องของรถเมล์ทางด่วน
21. สุขุมวิท 77 (อ่อนนุช) ถือเป็นถนนอ่อนนุชช่วงต้น โดยเริ่มจาก สามแยกอ่อนนุช ไปจนถึงสี่แยกอ่อนนุช โดยจะมีรถเมล์ สาย 1013 วิ่งอยู่ด้วย แต่เนื่องจาก 1013 เป็นรถเมล์ที่วิ่งลักษณะ รถเมล์ประจำซอย (มีเลข 4 หลัก) จึงไม่ได้นำมาคิด
22. อ่อนนุช (ม.เสรี) เป็นถนนอ่อนนุชช่วงสั้น ๆ โดยเริ่มคิดจาก สี่แยกอ่อนนุช ซึ่งคือถนนอ่อนนุชตัดกับถนนศรีนครินทร์ ไปจนถึง สามแยกผาสุก
23. อ่อนนุช(ประเวศ) เป็นถนนอ่อนนุชที่นับจากสามแยกผาสุก ไปสิ้นสุดยังขอบผนที่ แต่ในความเป็นจริงจะถือว่าสิ้นสุดที่ สี่แยกประเวศ
24. ศรีนครินทร์ (ซีคอน) ศรีนครินทร์ช่วงซีคอนสแควร์นี้จะเริ่มจากสี่แยกอ่อนนุช ไปจนถึง สี่แยกอุดมสุข
25. แยกพระราม9-ศรีนครินทร์ แยกพระราม 9 ศรีนครินทร์มีลักษณะเป็น 4 แยก แต่เนื่องจากพระราม 9 ช่วงมอเตอร์เวย์ยังไม่มีรถเมลวิ่ง เราจึงมองสี่แยกนี้ให้อยู่ในรูปแบบของสามแยก คือตัดช่วงมอเตอร์เวย์ออก
26. สามแยกอ่อนนุช สามแยกอ่อนนุช จะเป็นเหมือนปากทางเข้าถนนสุขุมวิท 77 คืออยู่ในรูปแบบซอยหนึ่งของถนนสุขุมวิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

27. สี่แยกอ่อนนุช สี่แยกอ่อนนุช คือสี่แยกที่เกิดจากการที่ถนนสุขุมวิท77 (อ่อนนุช) ตัดกันกับถนนศรีนครินทร์
28. สามแยกผาสุก สามแยกผาสุก เป็นสามแยกที่เกิดจากการที่ถนนพัฒนาการ มาบรรจบเข้ากับถนนสุขุมวิท 77 หรืออ่อนนุช
29. แยกพระโขนง-พระราม 4 เป็น 4 แยกที่เกิดจากการรวมตัวกัน ของแยกพระโขนง คือ ซอยสุขุมวิท 71 ของถนนสุขุมวิท รวมตัวกับสามแยกพระโขนงซึ่งเป็นจุดที่ถนนพระราม 4 มาบรรจบกับถนนสุขุมวิท โดยเนื่องจาก 2 สามแยกนี้อยู่ใกล้กันมาก และไม่มีป้ายรถเมล์ระหว่างกลาง 3 แยกนี้ จึงยุบรวมเป็น 4 แยกเดียว เพื่อความสะดวก
30. แยกเอกมัย-เพชรบุรี เป็นแยกที่เกิดจากถนนสุขุมวิท 63 เอกมัย ตัดกับถนนเพชรบุรี ซึ่งความจริงเป็น 4 แยก แต่เนื่องจากช่วงถนนเอกมัย-รามอินทรา นั้นมีรถเมล์ผ่านเพียงสายเดียว คือสาย 154 และเป็นสายที่มีรถน้อยมาก จึงตัดทิ้งไม่ได้นำมาสนใจ จึงทำให้สี่แยกนี้เราสนใจเพียง 3 แยก คือ ถนนเพชรบุรีตัดใหม่ทั้ง 2 ด้าน และ ถนน เอกมัย
31. สุขุมวิท(วัดธาตุทอง)เป็นช่วงหนึ่งของถนนสุขุมวิทเริ่มจากแยกเอกมัยสุขุมวิท ไปจนถึงแยกพระโขนง
32. แยกเอกมัย-สุขุมวิท เป็น 4 แยกที่เกิดจาก 3 แยก ที่อยู่ใกล้ ๆ กัน 2 อันรวมกันคือแยกเอกมัยสุขุมวิท เกิดจากปากซอยสุขุมวิท 63 แยกออกจากถนนสุขุมวิท และแยกกล้วยน้ำไทสุขุมวิท เกิดจากปากซอยสุขุมวิท 42 แยกออกจากถนนสุขุมวิท เมื่อเรานำสามแยกทั้ง 2 มารวมกัน เราก็จะมองแยกนี้ให้อยู่ในรูปของสี่แยก สี่แยกหนึ่ง
33. สุขุมวิท 42 (กล้วยน้ำไท)เป็นซอยที่เกิดจาก ถนนสุขุมวิท เชื่อมต่อไปยังถนนพระราม4 คือตั้งแต่แยกเอกมัย สุขุมวิท ไปยังแยกกล้วยน้ำไท เรามีการสมมติให้เป็นอู่ของสาย 71 ,98 และสาย 133
34. พระราม 4 (พระโขนง) เป็นถนนพระราม 4 ช่วงท้าย โดยนับเริ่มจาก แยกกล้วยน้ำไทไปยังแยกพระโขนงพระราม 4
35. แยกกล้วยน้ำไทเป็นสี่แยกที่เกิดจากการบรรจบกันของถนนสุขุมวิท42 กล้วยน้ำไท กับถนนอาจณรงค์ บรรจบกับถนนพระราม 4 ทำให้เกิดเป็น 4 แยก
36. สุขุมวิท 63 (เอกมัย) เป็น ถนนสุขุมวิท ที่แยกออกไปเชื่อมต่อกับถนนเพชรบุรีตัดใหม่ เริ่มจากแยกเอกมัยสุขุมวิท ไปยัง แยกเอกมัย เพชรบุรี
37. อาจณรงค์ ถนนอาจณรงค์ ที่เราสนใจในแผนที่ จะนับเริ่มตั้งแต่ แยกกล้วยน้ำไท ไปยังจุดสิ้นสุดของแผนที่
38. เพชรบุรีตัดใหม่ (RCA) เป็นถนนเพชรบุรีตัดใหม่ช่วงตั้งแต่แยกอโศกเพชรบุรี ไปจนถึงแยกเอกมัยเพชรบุรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

39. สุขุมวิท (สวนเบญจศิริ) เป็นถนนสุขุมวิทช่วงตั้งแต่แยกอโศกสุขุมวิท ไปจนถึงแยกเอกมัย-สุขุมวิท
40. พระราม 4 (คลองเตย) เป็นถนนพระราม 4 ช่วงตั้งแต่แยกคลองเตย ไปจนถึงกล้วยน้ำไท ระหว่างถนนจะมีแยก อื่น ๆ ด้วย แต่เราไม่ได้นำมาสนใจ
41. รัชดา (ฟอร์จูน) เป็นถนนรัชดาภิเษกช่วงตั้งแต่เริ่มแผนที่ จนสิ้นสุดที่สี่แยกอ.ส.ม.ท. ในความเป็นจริงจะเริ่มจากถนนประชาอุทิศ
42. ดินแดง ถ้าดูจากในแผนที่จะเห็นขอบเขตไม่ครบ คือเห็นจากสุดแผนที่ไปยัง สี่แยก อ.ส.ม.ท. แต่ในความเป็นจริงจะคลอบคลุมตั้งแต่ แยกดินแดง ไปจนถึง สี่แยก อ.ส.ม.ท.
43. สี่แยก อ.ส.ม.ท. เป็นการบรรจบกันบนถนนรัชดาภิเษก ของถนน 2 ฝั่งในแนวนอนคือ ถนนพระราม 9 และถนนดินแดง
44. อโศก-ดินแดง ถนนอโศก ดินแดง เป็นถนนรัชดาภิเษกช่วงหนึ่ง เริ่มตั้งแต่ สี่แยก อ.ส.ม.ท. มาสิ้นสุดที่สี่แยกอโศกเพชรบุรี
45. เพชรบุรี (มักกะสัน)เป็นถนนเพชรบุรีช่วงแรกที่เราสนใจ โดยเริ่มจากสุดแผนที่มาสิ้นสุดที่สี่แยกอโศกเพชรบุรี ซึ่งสุดของแผนที่นั้น คือ แยกมักกะสัน
46. สี่แยกอโศกเพชรบุรี เป็นสี่แยกที่เกิดจากการตัดกันของถนนรัชดาภิเษก ช่วงอโศก กับถนนเพชรบุรีตัดใหม่
47. สุขุมวิท 21 (อโศก) เป็นถนนรัชดาภิเษก ช่วงหนึ่งที่เหมือนเป็นซอยของถนนสุขุมวิท เริ่มตั้งแต่แยกอโศกสุขุมวิทถึงแยกอโศกเพชรบุรี
48. สุขุมวิท (นานา) เป็นถนนสุขุมวิทช่วงแรกที่เราสนใจ โดยเริ่มจากสุดแผนที่มาสิ้นสุดที่สี่แยกอโศกสุขุมวิท ซึ่งสุดของแผนที่นั้น คือ แยกนานา
49. สี่แยกอโศกสุขุมวิท เป็นการตัดกันของถนนสองสาย คือถนนรัชดาภิเษก ตัดกับถนนสุขุมวิท จึงเหมือนถนนรัชดาภิเษกช่วงนี้เป็นซอยของถนนสุขุมวิท
50. รัชดา (ศูนย์ประชุม) เป็นช่วงหนึ่งของถนนรัชดาภิเษก ที่มีศูนย์ประชุมแห่งชาติสิริกิติ์ เริ่มจากแยกอโศกสุขุมวิท ไปสิ้นสุดที่แยกคลองเตย
51. พระราม 4 (สวนลุม) เป็น ถนนพระราม 4 ช่วงแรกที่เราสนใจ โดยเริ่มจากสุดแผนที่มาสิ้นสุดที่ แยกคลองเตย ซึ่งสุดของแผนที่ที่เราไม่สามารถแสดงช่วงถนนได้ต่อนั้นคือใน่ส่วนของ บริเวณ บ่อนไก่ สวนลุม และแยกสาธร
52. แยกคลองเตย ถ้าดูจากแผนที่จะเป็น 4 แยก ที่ตัดกันระหว่างถนนรัชดาภิเษก กับ ถนนพระราม 4 แต่เรานำเอาช่วงถนนสุนทรโกษามารวมกับ 4 แยกนี้ด้วยด้วย จึงทำให้เกิดเป็น 5 แยก ขึ้นมาในมุมมองใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

53. พระราม 3 เป็นถนนรัชดาภิเษกช่วงหนึ่ง ที่มีชื่อเรียกใหม่ โดยจะเริ่มจากแยกคลองเตย ไปยังจุดสิ้นสุดของแผนที่ ในความเป็นจริงแล้วจะต่อเนื่องไปจนถึง บริเวณแยกถนนนราธิวาสราชนครินทร์
54. สุนทรโกษา เป็นถนนที่นำมาพ่วงรวมเป็นส่วนหนึ่งของแยกคลองเตย โดยถนนสายนี้จะรองรับทางที่ไปยังอู่รถเมล์จำนวนมาก ทั้งที่เป็น อู่ท่าเรือคลองเตย อู่กรมศุลกากร และอู่คลองเตย จึงต้องนำมาคิดด้วย
55. แยกอุดมสุข เป็นถนนช่วงสุขุมวิท 103 ตัดกับ ถนนศรีนครินทร์ ในความเป็นจริงแล้วแยกนี้จะมีลักษณะเป็น 4 แยก แต่เนื่องจากว่าถนนอุดมสุขช่วงแรกนั้นไม่มีรถเมล์ที่เป็นรถใหญ่วิ่งมีเฉพาะรถเมล์ที่วิ่งในซอยซึ่งเราไม่ได้นำมาคิดด้วย จึงทำให้แยกดังกล่าวมีลักษณะเป็น 3 แยกแทน คือไม่นับส่วนถนนอุดมสุขช่วงแรก คิดเฉพาะ ถนนศรีนครินทร์ และถนนอุดมสุขช่วงหลัง
56. สุขุมวิท 103 (สวนหลวงร.9) เป็นช่วงถนนที่มีระยะจากจุดที่สิ้นสุดในแผนที่ ไปยังแยกอุดมสุข แต่ในความเป็นจริงจะสิ้นสุดเพียงแค่ถึงอู่รถเมล์สาย 40และ 206 เท่านั้นเนื่องจากช่วงถัดจากนั้นจะไม่มีรถเมล์วิ่งผ่าน
57. ศรีนครินทร์ (วัดศรีเอี่ยม) เป็นศรีนครินทร์ช่วงที่นับตั้งแต่ แยกอุดมสุข มาจนถึงสี่แยกบางนา กม. 4
58. แยกบางนาสุขุมวิท ถ้าดูจากแผนที่จะพบว่าเป็น 4 แยก ที่เกิดขึ้นจาก ถนนบางนา และถนนสรรพาวุธ มาบรรจบกับถนนสุขุมวิท แต่เนื่องจากถนนสรรพาวุธนั้นไม่มีรถเมล์วิ่งผ่าน เราจึงไม่ได้นำมาคิดคำนวณด้วย เพราะฉะนั้นจะเหลือเพียง ถนนสุขุมวิททั้ง 2 ช่วง และถนนบางนาที่สนใจเท่านั้น
59. บางนา (กม.1 - 4) เป็นถนนบางนา-ตราด ช่วงแรกของถนน โดยนับเริ่มตั้งแต่ กม.ที่ 1 บริเวณแยกบางนาสุขุมวิท ไปจนถึงกม. 4 ที่บริเวณแยกบางนา กม.4 ที่เป็นการตัดกันของถนนบางนา กับ ถนนศรีนครินทร์
60. แยกบางนา กม.4 ตามที่ได้กล่าวมาแล้วนั้น จะพบว่าแยกบางนา กม.4 นี้จะเกิดจากการตัดกันของถนนบางนา-ตราด และถนนศรีนครินทร์
61. บางนา (ราม 2) เป็นถนนบางนา-ตราด ช่วงสุดท้ายที่เราสนใจ โดยจะเริ่มจากแยกบางนา กม. 4 ไปจนถึง มหาวิทยาลัย รามคำแหง วิทยาเขต บางนา หรือที่เรียกว่า ราม 2 จะสังเกตเห็นว่าเป็นจุดที่อยู่ปลายสุดของแผนที่ จะเห็นเพียงบางส่วน
62. สุขุมวิท (สำโรง) เป็นถนนสุขุมวิท ช่วงท้าย ๆ ที่เริ่มเข้าสู่ จังหวัดสมุทรปราการ จะมีอาณาเขต จาก แยกบางนาสุขุมวิท ไปจนถึงสุดแผนที่ แต่ในความเป็นจริงจะยาวต่อเนื่องไปถึงบริเวณ สำโรง คือช่วงถนนสุขุมวิท ตัดกับถนนเทพารักษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

63. ศรีนครินทร์(เทพารักษ์) เป็นโหนดสุดท้าย โดยเป็นโหนดที่แสดง ถนนศรีนครินทร์ ช่วงท้าย ๆ ที่เริ่มเข้าสู่ จังหวัดสมุทรปราการ จะมีอาณาเขต จาก แยกบางนากม. 4 ไปจนถึงสุดแผนที่ แต่ในความเป็นจริงจะยาวต่อเนื่องไปถึงบริเวณ เทพารักษ์ คือช่วงถนนศรีนครินทร์ตัดกับถนนเทพารักษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


[1] Stuart J.Russell and Peter Norvig : "*Artificial Intelligence : A Modern Approach*", Prentice-Hall International 1995

[2] Sartaj Sahni : "*Data Structures, Algorithms, and Applications in Java* ", McGraw-Hill International Edition 2000

[3] วีระศักดิ์ ซึงถาวร,ดร. : "*Fundamental of Java Programming Volume I* ", Sum Publishing 1998

[4] วีระศักดิ์ ซึงถาวร,ดร. : "*Fundamental of Java Programming Volume II* ", SE-ED Ucation 2000

[5] Department of Computer Science University of Saskatchewan : "*Tutorials and Applets for Teaching Basic Computer Science Concepts* "
http://www.cs.usask.ca/research/research_groups/aries/projects/applets/tutorials/ ,1999


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้