พจนานุกรม ไทย-อังกฤษ-พม่า

THAI-ENGLISH-BURMESE DICTIONARY

พีระเดช ลิ้มวงศ์
PEERADECH LIMWONG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
พิมพ์ครั้งที่ 1
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2553

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

พจนานุกรม ไทย – อังกฤษ – พม่า

THAI – ENGLISH – BURMESE DICTIONARY

พีรเดช ลิ้มวงศ์

PEERADECH LIMWONG

เลขหมู่.....
เลขทะเบียน.....74826.....
วัน,เดือน,ปี.....11 ต.ค. 2550.....

.b.....11 8 29901.....
.i.....

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2550

THAI – ENGLISH – BURMESE DICTIONARY

PEERADECH LIMWONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN INFORMATION ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2007

COPYRIGHT 2007

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยคำแนะนำ คำปรึกษา และแนวความคิดจาก รศ. ดร. ชวลิต เบญจางคประเสริฐ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ที่ได้ให้การแนะนำและสนับสนุนทางด้านข้อมูลต่าง ๆ ที่เป็นประโยชน์ที่ใช้ในการดำเนินงานวิจัย ให้ตลอดเวลาเป็นสิ่งที่ผู้วิจัยขอขอบพระคุณท่านมา ณ ที่นี้

ขอขอบพระคุณ รศ. นภพินทุ์ อนันตรศิริชัย รศ. อรลาภ แสงอรุณ และ ดร. พิทักษ์ ธรรมวาริน ที่ได้ให้กำลังใจ และ มอบคำแนะนำที่ดีให้ตลอดเวลา

ขอขอบคุณ พ.อ. เอกชาติ วงศ์ดี พ.อ. ศุภกร เทอดสุวรรณและ พ.ท. รณกร แพทย์ประสิทธิ์ ที่ได้ให้โอกาส ในการทำวิทยานิพนธ์ฉบับนี้จนสำเร็จ

ขอขอบคุณ ร.อ. หญิง ศิริพรรณ จินดาวงษ์ รน. ที่ได้ให้คำแนะนำด้านภาษาพม่า และพิสูจน์ตัวอักษรจนวิทยานิพนธ์ ได้เป็นฉบับที่สมบูรณ์

ขอขอบคุณ คุณ สุนิสา คุณารักษ์ และ หยาดพิรุณ เพิ่มสุขจิต ที่ได้ให้กำลังใจ และมอบคำแนะนำที่ดี ให้ตลอดเวลา

และขอกราบขอบพระคุณ คุณพ่อ สัก ลิ้มวงศ์ คุณแม่เทิม ลิ้มวงศ์ และ ครอบครัวอันเป็นที่รักยิ่ง ที่ได้ให้การสนับสนุน และ ส่งเสริมทางด้านการศึกษา รวมถึงกำลังใจ และความช่วยเหลือใน ทุก ๆ ด้าน ตลอดเวลา

ขอขอบคุณ น้อง ๆ ภาควิชาวิศวกรรมสารสนเทศ ที่ให้คำแนะนำ และความคิดเห็น ตลอดจนถึงความช่วยเหลือในทุก ๆ ด้าน และกำลังใจเสมอมา

พันโท พีรเดช ลิ้มวงศ์

สารบัญ

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ..... IV

สารบัญรูป.....VII

สารบัญตาราง..... IX

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 หลักการใหม่ที่นำเสนอ.....2

1.3 การเปรียบเทียบหลักการที่มีอยู่เดิม.....2

บทที่ 2 โครงสร้างข้อมูล.....4

2.1 หลักการเบื้องต้นของโครงสร้างข้อมูล.....4

2.2 การค้นหาข้อมูล (Searching).....5

2.2.1 การค้นหาแบบไบล์ด (Blind Search).....5

2.2.2 การค้นหาแบบฮิวริสติก(Heuristic Search).....6

2.3 การค้นหาดีสุดก่อนจนถึงลำดับที่ n (n-Best-first search).....7

2.4 โครงสร้างข้อมูลแบบ Lattice.....9

2.4.1 คำนิยามและหลักการที่เกี่ยวข้อง.....9

2.4.2 โครงสร้างข้อมูลและอัลกอริทึมแบบ Lattice.....10

2.5 การค้นหาข้อมูลแบบลิเนียร์เซิร์ท(LinearSearch).....13

2.6 การค้นหาข้อมูลไบนารีทรี (Binary Trees).....15

2.7 ไบนารีทรีแบบสมบูรณ์(Complete binary tree) 16

2.8 ไบนารีทรีแบบขยาย : 2-ทรี (Extended Binary Trees : 2 – Trees) 17

2.9 การเก็บไบนารีทรีแบบลิงค์ลิสต์.....18

2.10 อัลกอริทึมในการเข้าถึงข้อมูลแบบใช้สแตก.....20

2.10.1 การเข้าถึงข้อมูลแบบพรีออร์เดอร์.....20

2.10.2 การเข้าถึงข้อมูลแบบอินออร์เดอร์.....21

สารบัญ(ต่อ)

หน้า

2.10.3 การเข้าถึงข้อมูลแบบโพสท์ออร์เดอร์....23

2.11 ไบนารีเซิร์ททรี (Binary search trees) 24

2.11.1.การค้นหาและการแทรกในไบนารีเซิรท์ทรี....24

2.11.2.ความซับซ้อนของอัลกอริทึมการค้นหา....25

2.11.3.การประยุกต์ใช้งานไบนารีเซิรท์ทรี....25

2.11.4.การลบโหนดออกจากไบนารีเซิรท์ทรี....25

บทที่ 3 การค้นหาข้อมูลไบนารีเซิร์ททรี....26

3.1 ตัวอย่างโครงสร้างต้นไม้ของคำต่างๆ....28

3.2 ความเร็วในการสืบค้น....31

บทที่ 4 สถาปัตยกรรมของระบบ....32

4.1 ข้อมูลในระบบ....32

4.2 ข้อมูลที่ใช้ในสถาปัตยกรรมของระบบ....33

4.3 ระบบดิกชันนารี....34

บทที่ 5 ผลการทดลอง....38

5.1 วิธีดำเนินการทดลอง....38

5.2 ผลการทดลอง....49

5.3 การแสดงผลเมื่อคำศัพท์นั้นมีภาพ 45

5.4 การแสดงผลเมื่อคำศัพท์นั้นไม่มีแต่จะแสดงคำศัพท์ใกล้เคียง....49

5.5 การแสดงผลเมื่อคำศัพท์นั้นไม่มีภาพ....56

5.6 ผลการทดลองของคำศัพท์อื่น ๆ....60

บทที่ 6 สรุปผลการวิจัย....62

บรรณานุกรม....63

ภาคผนวก....65

ภาคผนวก ก โปรแกรม....66

ภาคผนวก ข ตัวอย่างฐานข้อมูล 3 ภาษา....74

สารบัญ(ต่อ)

หน้า

ภาคผนวก ค ผลงานวิจัยที่ได้รับการตีพิมพ์.....139

ประวัติผู้เขียน.....140

สารบัญรูป

รูปที่ หน้า

2.1 การจัดเรียงเป็นโมดูลย่อย....4
2.2 ขั้นตอนของการค้นหาแบบดีที่สุดก่อน....7
2.3 การค้นหาแบบดีสุดก่อน....8
2.4 โครงสร้างข้อมูล FC....10
2.5 ตัวอย่างฐานข้อมูลและ FCC แต่ละลำดับ....12
2.6 โครงสร้างของไบนารีทรี....17
2.7 ทรีที่สมบูรณ์ของ T_{26}....18
2.8 การแปลงไบนารีทรี T เป็น 2-ทรี....19
2.9 ลำดับการเข้าถึงข้อมูล....20
2.10 อัลกอริทึมในการเข้าถึงข้อมูลแบบใช้สแตก....21
2.11 การเข้าถึงข้อมูลแบบพรีออร์เดอร์....22
3.1 ลำดับขั้นตอนในการทำงานของนอร์มอลไลซ์ไบนารีเซิร์ททรี....26
3.2 โครงสร้างต้นไม้ของคำว่า " พยาบาล "....27
3.3 โครงสร้างต้นไม้ของคำว่า " cobra "....29
3.4 โครงสร้างต้นไม้ของคำว่า " quartermaster "....30
4.1 Context diagram....34
4.2 Data flow diagram level 1 ของระบบ....34
4.3 การเพิ่มและลดคำศัพท์....35
4.4 การค้นหาคำศัพท์....36
5.1 อัตราความเร็วการค้นหาคำศัพท์ทั้งหมด 10,000 คำ....42
5.2 อัตราความเร็วการค้นหาคำศัพท์ทั้งหมด 15,000 คำ....43
5.3 อัตราความเร็วการค้นหาคำศัพท์ทั้งหมด 20,000 คำ....44
5.4 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า " Lake "....45
5.5 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า " Satellite "....46
5.6 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า " คอมพิวเตอร์ "....47
5.7 ภาพเมื่อใส่คำศัพท์เป็นภาษาพม่าคำว่า " နာလန္ဓ "....48
5.8 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า " handoff "....49
5.9 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า " handoff "....50

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.10 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทยคำว่า “แก้วทดลอง”.....51
5.11 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทยคำว่า “แก้วทดลอง”.....52
5.12 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทยคำว่า “แก้วทดลอง”.....53
5.13 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษคำว่า “penicillium”.....54
5.14 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษคำว่า “penicillium”.....55
5.15 อินพุต/เอาต์พุตคำว่า “อากาศ”.....56
5.16 อินพุต/เอาต์พุตคำว่า “smell”.....57
5.17 อินพุต/เอาต์พุตคำว่า “energy” 58
5.18 อินพุต/เอาต์พุตคำว่า “ไฟ”.....59

สารบัญตาราง

ตารางที่ หน้า

5.1 อัตราความเร็วการค้นหาคำศัพท์ทั้งหมด 10,000 คำ.....39

5.2 อัตราความเร็วการค้นหาคำศัพท์ทั้งหมด 15,000 คำ.....40

5.3 อัตราความเร็วการค้นหาคำศัพท์ทั้งหมด 20,000 คำ.....41

บทที่ 1
บทนำ

1.1 ความเป็นมา และความสำคัญของปัญหา

เนื่องจากในปัจจุบันนี้โลกได้ก้าวไปข้างหน้าอย่างรวดเร็วมาก เหตุผลที่เป็นปัจจัยอย่างหนึ่งเพราะ มีเทคโนโลยีที่ทันสมัยรองรับกับความต้องการของผู้ใช้ ดังนั้นการติดต่อสื่อสาร จึงเข้ามามีบทบาทและมีความสำคัญเป็นอย่างมาก ไม่ใช่เพียงการติดต่อสื่อสารทางด้านธุรกิจเท่านั้น แต่ยังหมายรวมถึงการติดต่อสื่อสารทางด้านการทหารด้วย และ สำหรับการติดต่อสื่อสารที่ได้กล่าวถึงมานี้ ก็ไม่ได้หยุดอยู่แค่การติดต่อ หรือการใช้ที่เป็นภาษาไทยเพียงเท่านั้น แต่ยังหมายรวมถึงภาษาอื่น ๆ ที่เราจำเป็นต้องพบเจออย่างหลีกเลี่ยงไม่ได้ด้วย และ ที่สำคัญคือไม่สามารถเข้าใจความหมายของภาษาอื่น ๆ ได้ทุกถ้อยคำเพราะฉะนั้นจึงจำเป็น ที่จะต้องมีเครื่องมือเข้ามาช่วยในการหาความหมายของคำแต่ละคำให้ได้อย่างถูกต้อง นอกเหนือจากนี้แล้วเวลาที่ใช้ในการหาความหมายของคำแต่ละคำหรือศัพท์แต่ละตัวนั้นยังจะต้องใช้เวลาที่น้อยที่สุดด้วย

จากที่ผ่านมามีผู้คิดค้นอัลกอริทึม (Algorithm) หรือวิธีต่าง ๆ ขึ้นมาอยู่ด้วย หลายวิธีอย่างเช่น การนำเอาวิธีการค้นหาข้อมูลแบบ N-best [1-3,8] มาใช้ แต่วิธีนี้มีข้อจำกัดอยู่ที่ว่าข้อมูล ที่ใช้จะต้องมีขนาดที่ใหญ่มากพอ ส่วนวิธีของ Lattice [12] เป็นวิธีที่มีการทำงานร่วมกับวิธีของ N-best เพื่อลดขนาดของข้อมูล แต่วิธีนี้ก็ยังคงมีข้อจำกัดอยู่ที่เวลาที่ใช้ในการประมวลผลที่ค่อนข้างสูงมาก ส่งผลให้ในบางครั้งเกิดความผิดพลาดในการประมวลผลขึ้น

ดังนั้นในวิทยานิพนธ์ฉบับนี้ จึงนำเสนออัลกอริทึมที่ใช้เวลาในการประมวลผลหรือการค้นหาความหมายของคำหรือคำศัพท์ที่ต้องการ โดยใช้เวลาน้อยกว่าวิธีการของ N-best และ Lattice เพื่อเอาไว้ใช้สำหรับการออกแบบเครื่องมือที่เอาไว้ช่วยในการหาความหมายของคำศัพท์ก็คือ ดิกชันนารี (Dictionary) นั่นเอง

สำหรับการสร้างอัลกอริทึมนั้นคือการสร้างหรือออกแบบแนวคิดอย่างมีเหตุผลที่ ผู้เขียนโปรแกรมใช้ในการอธิบายวิธีการอย่างเป็นขั้นตอนตามลำดับ ในการที่จะพัฒนาโปรแกรมนั้น ๆ ให้กับผู้ที่สนใจได้ทราบถึงขั้นตอนต่างๆ ในการเขียนหรือพัฒนาตัวโปรแกรม ขณะเดียวกันก็ช่วยให้ผู้สนใจได้ตรวจสอบขั้นตอนต่าง ๆ ในการทำงานและความถูกต้องในแต่ละขั้นตอน โดยผู้ที่สนใจไม่จำเป็นต้องเขียนโปรแกรมเป็นก็ได้ สำหรับองค์ประกอบของการจัดทำอัลกอริทึมมีดังต่อไปนี้

1.1.1 การวิเคราะห์ (Analysis)

1.1.2 การออกแบบ (Design)

1.1.3 การเขียนโปรแกรม (Coding/Programming)

1.1.4 การทดสอบและแก้ไขข้อผิดพลาดของโปรแกรม (Testing and debugging)

1.1.5 การจัดทำเอกสารและการบำรุงรักษา (Documentation and maintenance)

โดยองค์ประกอบทั้ง5ส่วนนี้สามารถจะนำเอามาประยุกต์ใช้ได้ทั้งกับ โปรแกรมที่มีระบบงานคอมพิวเตอร์ขนาดเล็กและขนาดใหญ่ได้เพราะฉะนั้นการจัดเก็บข้อมูลและการกระทำกับ โครงสร้างข้อมูลนั้นถือได้ว่าเป็นใจความสำคัญหรือเป็นหัวใจหลักของวิทยานิพนธ์ฉบับนี้ดังนั้นในลำดับต่อไปจะขอกล่าวถึงรายละเอียดในแต่ละส่วนของอัลกอริทึมต่าง ๆ ที่ได้มีการนำเสนอในวิทยานิพนธ์ฉบับนี้ตามลำดับ

1.2 หลักการใหม่ที่นำเสนอ

จากที่ได้กล่าวมาตั้งแต่ตอนต้นแล้วว่าหัวใจหลักที่สำคัญสำหรับวิทยานิพนธ์ฉบับนี้คือ การสร้างอัลกอริทึมซึ่งอัลกอริทึมนั้นนับได้ว่าเป็นขั้นตอนในการแก้ปัญหาต่าง ๆ ดังนั้นการพัฒนาอัลกอริทึมที่มีประสิทธิภาพในการจัดเก็บข้อมูล เวลา และ พื้นที่ จะเป็นสิ่งสำคัญที่แสดงให้เห็นถึงประสิทธิภาพของอัลกอริทึมและความซับซ้อนของอัลกอริทึมเป็นฟังก์ชันที่มีการกำหนดเวลา และพื้นที่ของการทำงานในเทอมของขนาดข้อมูลโดยอัลกอริทึมที่มีประสิทธิภาพไม่สามารถที่จะ ใช้กับโครงสร้างของข้อมูลได้ทั้งหมดเนื่องจากข้อมูลแต่ละส่วนมีความแตกต่างกัน ซึ่งการกระทำกับโครงสร้างข้อมูลที่มีเวลาและพื้นที่เข้ามาเกี่ยวข้องนั้น หากว่าเพิ่มพื้นที่การเก็บข้อมูลให้มากขึ้น เวลาอาจจะลดลงได้ โดยตัวอัลกอริทึมนั้น อาจจะมีความยุ่งยากและสลับซับซ้อนดังนั้นเราจึงต้องทำการแปลงตัวอัลกอริทึมให้ไปเป็นตัวโปรแกรมก่อน เพื่อ ทำให้การทำงานนั้นดูง่ายขึ้น และ ทำการแบ่งออกเป็นโมดูลย่อย ๆ สำหรับในส่วนของโปรแกรมนั้นจะประกอบไปด้วยส่วนของโมดูลหลักที่จะบรรจุรายละเอียดทั่วไปของอัลกอริทึมและมีการเรียกใช้โมดูลย่อยต่าง ๆ ซึ่งในโมดูลย่อยจะมีรายละเอียดมากกว่าโมดูลหลักหรืออาจมีการอ้างถึงโมดูลย่อยอื่น ๆ อีกสำหรับในส่วนการจัดโครงสร้างของโปรแกรมเป็นโมดูลย่อย ๆ นั้น จะต้องใช้พื้นฐานทางด้านผังโปรแกรมและโครงสร้างทางด้านลอจิกหลาย ๆ อย่างประกอบกัน

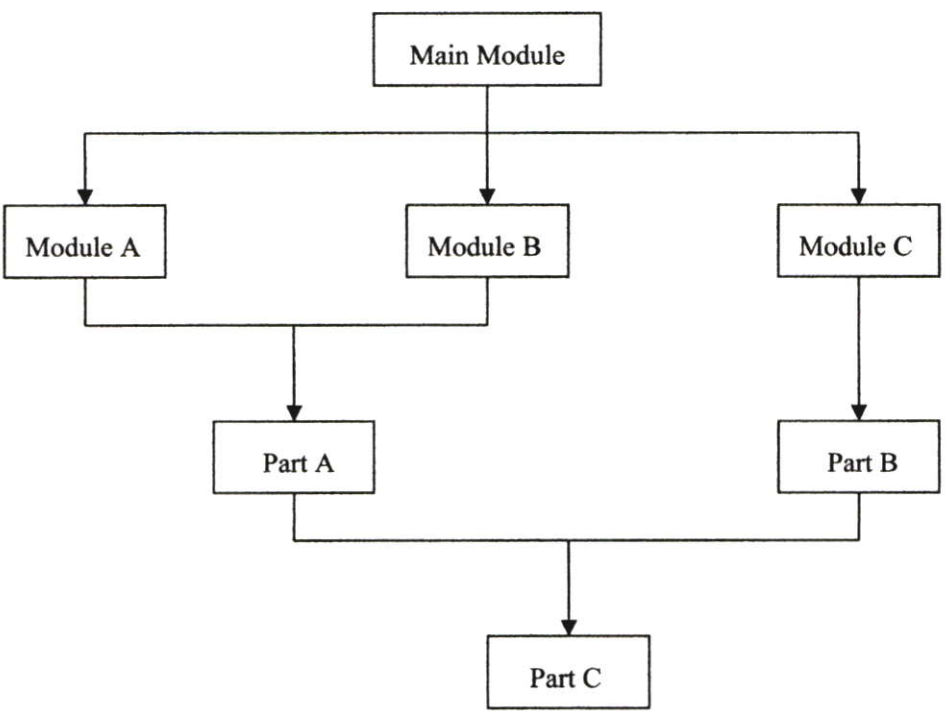
1.3 การเปรียบเทียบกับหลักการที่มีอยู่เดิม

สำหรับในส่วนของการออกแบบอัลกอริทึมที่ใช้ในวิทยานิพนธ์ฉบับนี้ เริ่มต้นจากการนำเอาข้อมูล ที่เก็บอยู่ในโปรแกรมอื่น อย่างเช่น ข้อมูลที่ถูกเก็บอยู่ในโปรแกรม Excel โหลดเข้ามายังส่วน Workspace ของโปรแกรม MATLAB เพื่อทำหน้าที่เป็นอินพุต (Input) ของระบบ สำหรับส่วนที่เป็นอินพุตนั้นสามารถเป็นไปได้ทั้งหมด 3 กรณี หรือ 3 ภาษา ได้แก่ ภาษาไทย ภาษาอังกฤษ และภาษาพม่า จากนั้นในลำดับต่อไป จะทำการเปรียบเทียบระหว่างข้อมูลที่เป็นอินพุตกับข้อมูลที่อยู่ในฐานข้อมูลที่มีอยู่ ถ้าอินพุตเป็นภาษาไทยก็จะทำการเปรียบเทียบกับฐานข้อมูลที่เป็นภาษาไทย หรือถ้าอินพุตเป็นภาษาอังกฤษก็จะทำการเปรียบเทียบกับฐานข้อมูลที่เป็นภาษาอังกฤษ หรือถ้าอินพุทเป็นภาษาพม่าก็จะทำการเปรียบเทียบกับฐานข้อมูลที่เป็นภาษาพม่า เป็นต้น เมื่อเข้าไปยังฐานข้อมูล

ในภาษานั้น ๆ แล้วจะใช้การแบ่งกิ่งหรือลำดับเป็นตัวเปรียบเทียบ ในการค้นหาข้อมูลจากค่า เทรสโฮล (Threshold) ซึ่งค่าเทรสโฮลที่นำมาใช้ในการพิจารณานั้นกำหนดเอาไว้ให้มีค่าเท่ากับ 7 กล่าวคือ เมื่อค่าความยาวของอินพุตมีค่ามากกว่า 7 ให้กระบวนการค้นหาเริ่มต้นจากทางด้านขวาของทรี (Tree) หรือต้นไม้ แต่ถ้าค่าความยาวของอินพุตมีค่าน้อยกว่า 7 จะให้กระบวนการค้นหาเริ่มต้นจากทางด้านซ้ายของต้นไม้แทนนอกจากนี้ยังสามารถแสดงความหมายของคำศัพท์เป็นภาษาไทย ภาษาอังกฤษ หรือ ภาษาพม่า ได้แล้วนั้นยังสามารถแสดงเป็นรูปภาพและเสียงได้ด้วย ในกรณีของเสียงนั้นมีเฉพาะในภาษาพม่าเท่านั้น โดยอัลกอริทึมที่ใช้และลำดับขั้นตอนในการทำงานสามารถแสดงได้ดังในบทที่ 3

บทที่ 2
โครงสร้างข้อมูล

2.1 หลักการเบื้องต้นของโครงสร้างข้อมูล



รูปที่ 2.1 การจัดเรียงเป็นโมดูลย่อย

ในส่วนของการจัดแบ่งข้อมูลซึ่งสามารถทำได้หลายวิธีที่แตกต่างกันออกไป สำหรับการจัดโครงสร้างเชิงคณิตศาสตร์หรือลอจิกของข้อมูลนั้น เรียกว่า โครงสร้างข้อมูล (Data structures) โดยการเลือกรูปแบบในการจัดเก็บข้อมูล มีสิ่งที่จะต้องพิจารณาอยู่ด้วยกัน 2 ประการคือ

2.1.1 ต้อง มีโครงสร้าง ที่สมบูรณ์เพียงพอที่จะแสดงให้เห็นถึงความสัมพันธ์ของข้อมูลปัจจุบัน

2.1.2 โครงสร้าง ต้องง่ายต่อการดำเนินการสำหรับการดำเนินการกับโครงสร้างข้อมูลต่าง ๆ นั้น ขึ้นอยู่กับลักษณะของการใช้งาน ในส่วนนี้จะเป็นการกล่าวถึงวิธีการดำเนินการกับข้อมูลที่มีการใช้งานกันมาก 4 อย่างคือ

2.1.2.1 การเข้าถึงเรคคอร์ด (Traversing) เป็นการเข้าไปติดต่อกับเรคคอร์ดที่ต้องการ เพื่อทำการ เปลี่ยนแปลงหรือแก้ไขข้อมูลในเรคคอร์ดนั้น

2.1.2.2 การค้นหา (Searching) การค้นหาตำแหน่งของเรคคอร์ดตามค่าที่กำหนด เพื่อทำการเปลี่ยน แปลงหรือแก้ไขข้อมูลตามเงื่อนไข

2.1.2.3 การเพิ่ม (Inserting) เป็นการเพิ่มข้อมูลในเรคคอร์ดใหม่

2.1.2.4 การลบข้อมูลออก (Deleting)

ในบางกรณีอาจมีการจัดเก็บข้อมูลมากกว่า 1 อย่าง เช่น หากต้องการลบข้อมูลของเรคคอร์ดตามคีย์ที่กำหนดจะต้องทำการค้นหาเรคคอร์ดที่มีข้อมูลตรงกับคีย์ก่อนแล้ว จึงลบข้อมูล นอกจากนี้ยังมีการจัดการกับข้อมูลอีก 2 อย่างที่ควรพิจารณาคือ

1. การเรียงข้อมูล (Sorting) เป็นการจัดเรียงเรคคอร์ดตามค่าที่กำหนด เช่น การเรียงข้อมูลจากมากไปน้อยหรือเรียงตามตัวอักษร โดยมีฟิลด์ที่กำหนดเป็นคีย์

2. การรวมข้อมูล (Merging) เป็นการนำเอาข้อมูลหลายเรคคอร์ดที่อยู่คนละไฟล์และมีการเรียงข้อมูลแล้วมารวมกันเป็นไฟล์ที่มีการเรียงข้อมูลในไฟล์เดียวกัน

2.2 การค้นหาข้อมูล (Searching)

การค้นหาคำตอบ หรือการค้นหาข้อมูลในทางคอมพิวเตอร์ มักจะกระทำบนโครงสร้างข้อมูลแบบต้นไม้และกราฟ ทั้งนี้เพราะโครงสร้างข้อมูลในลักษณะนี้ สามารถทำให้การค้นหาทำได้สะดวกและสามารถพลิกแพลงการค้นหาได้ง่าย ในความเป็นจริงแล้ว การค้นหาข้อมูลบางครั้งสามารถกระทำบนโครงสร้างข้อมูลชนิดอื่นก็ได้ เช่น อาเรย์ แสตก และคิว แต่การจัดข้อมูลในโครงสร้างเช่นนี้ มีข้อจำกัดในการค้นหาข้อมูลมาก การค้นหาทำได้แบบเรียงลำดับ (Sequential search) เท่านั้น ซึ่งใช้ได้กับข้อมูลที่มีขนาดเล็ก ดังนั้นในการค้นหาข้อมูลที่มีขนาดใหญ่ ก่อนการค้นหา หรือระหว่างการค้นหาข้อมูลที่จะถูกค้นจะต้องถูกจัดให้อยู่ในรูปแบบของต้นไม้ หรือ กราฟเท่านั้น การค้นหาข้อมูลบนโครงสร้างต้นไม้และกราฟสามารถจำแนกได้ 2 แบบคือ การค้นหาแบบไบล์ด (Blind search) และการค้นหาแบบฮิวริสติก (Heuristic search)

2.2.1 การค้นหาแบบไบล์ด (Blind search)

การค้นหาแบบไบล์ดเป็นการค้นหาแบบที่เดินทางจากโหนดหนึ่งไปยังอีกโหนดหนึ่ง โดยอาศัยทิศทางเป็นตัวกำหนดการค้นหา ไม่ต้องมีข้อมูลอะไรมาช่วยเสริมการตัดสินใจว่า จะเดินทางต่อไปอย่างไร หรือกล่าวอย่างง่าย ๆ คือ การจะหยิบข้อมูลใดมาช่วยในการค้นหาต่อไป ไม่ต้องอาศัยข้อมูลใด ๆ ทั้งสิ้น นอกจากทิศทางซึ่งเป็นรูปแบบตายตัว การค้นหาแบบไบล์ด สามารถแบ่งย่อยได้ดังนี้ คือ การค้นหาทั้งหมดและการค้นหาบางส่วน

- การค้นหาทั้งหมด (Exhaustive search) คือ การค้นหาทั้งหมดของปริภูมิสถานะ

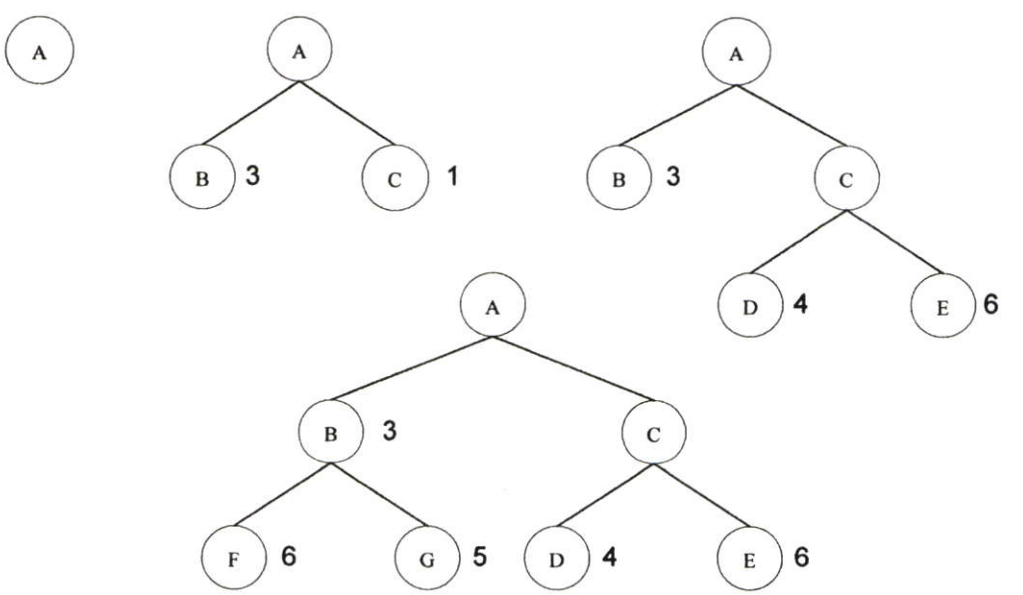
- การค้นหาบางส่วน (Partial search) การค้นหาเพียงบางส่วนของปริภูมิสถานะ ซึ่งในความเป็นจริงการค้นหาส่วนมากใช้การค้นหาเฉพาะบางส่วนเท่านั้น เนื่องจากปริภูมิสถานะมักมีขนาดใหญ่ ทำให้ไม่สามารถค้นหาได้ทั้งหมด ดังนั้นจึงมีความเป็นไปได้ว่าคำตอบที่ได้อาจไม่ใช่คำตอบที่ดีที่สุด การค้นหาแบบนี้สามารถแบ่งได้เป็น 2 ประเภทคือ การค้นหาแบบลึกก่อน (Depth first search) และการค้นหาแบบกว้างก่อน (Breadth first search)

2.2.2 การค้นหาแบบฮิวริสติก (Heuristic search)

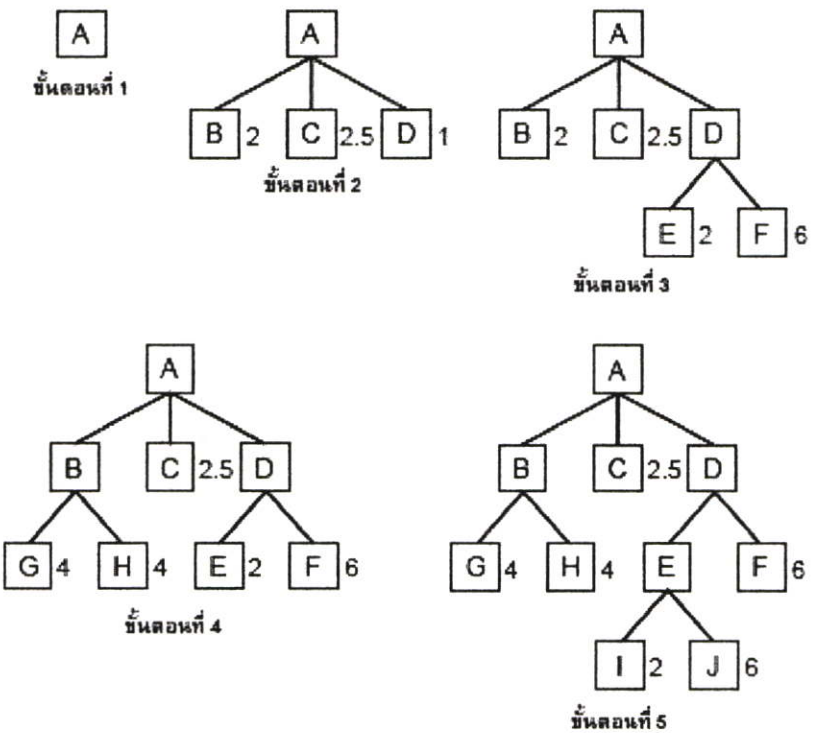
การค้นหาคำตอบอาศัยวิธีการทางฮิวริสติกมีความแตกต่างจากการค้นหาข้อมูล แบบธรรมดาและแบบฮิวริสติกนั้น อยู่ที่การค้นหาข้อมูลธรรมดา ผู้ที่ทำการค้นข้อมูลจะต้องตรวจสอบข้อมูลทีละตัวทุกตัวจนครบ แต่ฮิวริสติกจะไม่ลงไปดูข้อมูลทุกตัว วิธีการนี้จะเลือกได้คำตอบที่เหมาะสมให้กับการค้นหา ซึ่งมีข้อดี คือสามารถทำการค้นหาคำตอบจากข้อมูลที่มีขนาดใหญ่มาก ๆ ได้ แต่มีข้อเสีย คือคำตอบที่ได้เป็นเพียงคำตอบที่ดีเท่านั้น ไม่แน่ว่าจะดีที่สุด แต่เนื่องจากว่าปัญหาในบางลักษณะนั้นใหญ่มาก และเป็นไปไม่ได้ที่จะทำการค้นหาด้วยวิธีธรรมดา กระบวนการของฮิวริสติกจึงเป็นสิ่งที่จำเป็นในเรื่องของฮิวริสติก นั้นนอกจากจะมีการค้นหาแบบฮิวริสติกแล้ว ยังมีอีกสิ่งหนึ่งที่สำคัญ คือ ฮิวริสติกฟังก์ชัน(Heuristic function) ซึ่งหมายถึงฟังก์ชันที่ทำหน้าที่ในการวัดขนาดของความเป็นไปได้ในการแก้ปัญหา ซึ่งจะแสดงด้วยตัวเลข วิธีการดังกล่าวจะ กระทำได้โดยการพิจารณาถึงวิธีการ (Aspects) ต่าง ๆ ที่ใช้ในการแก้ปัญหา ณ สถานะหนึ่งว่าจะสามารถแก้ปัญหาได้ตามที่ต้องการหรือไม่โดยกำหนดเป็นน้ำหนักให้กับการแก้ปัญหาของแต่ละวิธี น้ำหนักเหล่านี้จะถูกแสดงด้วยตัวเลขที่กำกับไว้กับโหนดต่าง ๆ ในกระบวนการค้นหา และค่าเหล่านี้จะเป็นตัวที่ใช้ในการประมาณความเป็นไปได้ว่าเส้นทางที่ผ่านโหนดนั้นจะมี ความเป็นไปได้ในการนำไปสู่หนทางการแก้ปัญหาได้มากน้อยแค่ไหนจุดประสงค์ที่แท้จริงของฮิวริสติก ฟังก์ชันก็คือ การกำกับทิศทางของกระบวนการค้นหา เพื่อให้อยู่ในทิศทางที่ได้ประโยชน์สูงสุด โดยการบอกว่าควรเลือกเดินเส้นทางไหนก่อนในกรณีที่มีเส้นทางมากกว่าหนึ่งเส้นทางต้องเลือกกระบวนการค้นหาแบบฮิวริสติก โดยปกติแล้วจะต้องอาศัยฮิวริสติกฟังก์ชัน ทำให้การแก้ปัญหาหนึ่ง ๆ จะดีหรือไม่ ก็ขึ้นอยู่กับฮิวริสติกฟังก์ชัน ดังนั้นการค้นหาแบบนี้ จึงไม่มีอะไรเป็นหลักประกันว่าจะได้สิ่งที่ดีออกมา ด้วยเหตุนี้เองจึงเรียกการ ค้นหาแบบฮิวริสติกนี้ว่า Weak methods หรือจะกล่าวอีกนัยหนึ่ง คือ Weak methods เป็นกระบวนการควบคุมโดยทั่วไป (General-purpose control strategies) ซึ่งการค้นหาแบบนี้ สามารถแบ่งได้เป็น การค้นหาแบบปีนเขา (Hill climbing) การค้นหาแบบ Greedy (Greedy algorithm) และการค้นหาดีสุดก่อนจนถึงลำดับที่ n (N-best-first search)

2.3 การค้นหาดีสุดก่อนจนถึงลำดับที่ n (N-best-first search)[8]

เป็นกระบวนการค้นหาข้อมูลที่ได้นำเอาข้อดีของทั้งการค้นหาแบบลึกก่อน (Depth first search) และการค้นหาแบบกว้างก่อน (Breadth first search) มารวมกันเป็นวิธีการเดียว โดยที่แต่ละขั้นของการค้นหาในโหนดลูกนั้น การค้นหาแบบดีที่ดีก่อนจะเลือกเอาโหนดที่ดีที่สุด (Most promising) และการที่จะทราบว่าโหนดใดดีที่สุดนี้ สามารถทำได้โดยอาศัยฮิวริสติกฟังก์ชัน ซึ่งฮิวริสติกฟังก์ชันนี้ จะทำหน้าที่เหมือนตัววัดผล และให้ผลของการวัดนี้ออกมาเป็นคะแนน รูปที่ 2.2 เป็นตัวอย่างของการค้นหาแบบดีที่สุดก่อน ขั้นตอนนี้ เริ่มจากขั้นตอนที่ 1 สร้างโหนดราก (Root node) ในขั้นตอนที่ 2 สร้างโหนดลูก B และ C แล้วตรวจสอบโหนด B และ C ด้วยฮิวริสติกฟังก์ชัน ได้ผลออกมาเป็นคะแนนคือ 3 และ 1 ตามลำดับ จากนั้นให้เลือกโหนด C เป็นโหนดต่อไปที่สนใจ เพราะมีค่าน้อยกว่า (หมายเหตุ ในการเลือกนี้จะเลือกค่ามากสุด หรือน้อยสุดก็ได้ ขึ้นอยู่กับลักษณะของปัญหา) แล้วสร้างโหนดลูกให้กับโหนด C ในขั้นตอนที่ 3 ได้โหนด D และ E แล้วตรวจสอบคะแนนได้ 4 และ 6 ตามลำดับ จากนั้นทำการเปรียบเทียบค่าของโหนดท้ายสุด หรือเทอร์มินอลโหนด(Terminal node) ทุกโหนด ว่าโหนดใดมีค่าดีที่สุด ในที่นี้จะต้องเลือกโหนด B เพราะมีคะแนนเพียง 3 (เลือกคะแนนต่ำสุด) แล้วสร้างโหนดลูกตามขั้นตอนที่ 4 ได้ F และ G แล้วตรวจสอบคะแนนได้ 6 และ 5 คะแนน ตามลำดับ ทำเช่นนี้เรื่อย ๆ จนพบคำตอบหรือจนไม่สามารถสร้างโหนดต่อไปได้อีก



รูปที่ 2.2 ขั้นตอนของการค้นหาแบบดีที่สุดก่อน



รูปที่ 2.3 การค้นหาแบบดีสุดก่อน

2.3.1 อัลกอริทึม: การค้นหาแบบดีที่สุดก่อน

2.3.1.1 เริ่มด้วย OPEN ที่มีเพียงโหนดเริ่มต้น

2.3.1.2 ทำจนกว่าจะพบเป้าหมาย หรือว่าไม่มีโหนดเหลืออยู่ใน OPEN

2.3.1.2.1 เลือกโหนดที่ดีที่สุดใน OPEN

2.3.1.2.2 สร้างโหนดลูกให้กับโหนดที่ดีที่สุดนั้น

2.3.1.2.3 สำหรับโหนดลูกแต่ละตัวให้ทำดังต่อไปนี้

2.3.1.2.3.1 ถ้าโหนดนั้นยังไม่เคยถูกสร้างมาก่อนหน้านั้น ให้ตรวจสอบค่าของมันโดยใช้วิธีฮิวริสติกฟังก์ชัน แล้วเพิ่มเข้าไปใน OPEN แล้วบันทึกว่าเป็นโหนดแม่

2.3.1.2.3.2 ถ้าโหนดนั้นถูกสร้างมาก่อนหน้านี้แล้ว ให้เปลี่ยนโหนดแม่ของมัน ถ้าเส้นทางใหม่ที่ได้ดีกว่าโหนดแม่ตัวเดิม ในกรณีนี้ ให้ปรับเปลี่ยนค่าตามเส้นทางที่อาจจะเกิดขึ้น

2.4 โครงสร้างข้อมูลแบบ Lattice[12]

ในการหากฎความสัมพันธ์สามารถแบ่งออกเป็น 2 ขั้นตอน ได้แก่ ขั้นตอนที่ 1 เป็นการหา Frequent itemsets ทั้งหมด โดยหา itemsets ที่มีความถี่มากกว่า หรือ เท่ากับความถี่ขั้นต่ำ (Minimum support) ขั้นตอนที่ 2 เป็นการสร้างกฎความสัมพันธ์ โดยการนำ Freqeunt itemsets ซึ่งเป็นผลลัพธ์จากขั้นตอนที่ 1 มาเขียนให้อยู่ในรูป $X \Rightarrow Y$ โดยที่ X และ Y หมายถึง Itemset ใด ๆ ซึ่ง X และ Y จะต้องไม่มีสมาชิกซ้ำกัน และกฎดังกล่าวต้องมีค่าความเชื่อมั่น (Confidence) มากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ (Minimum confidence)

จะเห็นได้ว่าขั้นตอนที่ 2 ของการสร้างกฎความสัมพันธ์นั้นสามารถหาได้โดยตรงและไม่ยุ่งยากนัก ซึ่งต่างกับขั้นตอนที่ 1 ที่ต้องใช้ การนับ การเปรียบเทียบ การทำซ้ำ และการอ่านข้อมูลจากฐานข้อมูลขนาดใหญ่จำนวนหลายครั้ง ดังนั้น อัลกอริทึมโดยส่วนใหญ่ จึงเน้นการแก้ปัญหาขั้นตอนในการหา Frequent itemsets มากกว่าขั้นตอนการหากฎความสัมพันธ์ โดยที่อัลกอริทึมในช่วงแรก ๆ ที่เป็นที่รู้จักกันดีได้แก่ อัลกอริทึม Apriori อัลกอริทึม Mannila อัลกอริทึม Partition อัลกอริทึม Sampling และ อัลกอริทึม DIC (Dynamic Itemsets Counting) เป็นต้น ซึ่งอัลกอริทึมเหล่านี้มีพื้นฐานเดียวกันกับอัลกอริทึม Apriori โดยใช้คุณสมบัติพื้นฐาน ซึ่งมีหลักการอยู่ว่า " ถ้า itemsets ขนาด k ใด ๆ ไม่เป็น frequent แล้ว itemsets ขนาด k+1 ที่เป็น Superset ของ itemsets นั้นไม่เป็น frequent ด้วย" หรือกล่าวอีกนัยหนึ่งว่า " ถ้า itemsets ที่เป็น superset เป็น frequent แล้ว itemsets ที่เป็นสับเซต จะเป็น frequent ด้วย "

สำหรับคุณสมบัติพื้นฐานของอัลกอริทึมแบบ lattice ที่ต่างไปจากอัลกอริทึม Apriori จะใช้หลักการของอัลกอริทึม Close เป็นคุณสมบัติพื้นฐาน ซึ่งมีหลักการอยู่ว่า " ถ้า closed itemsets ขนาด k ใด ๆ ไม่เป็น frequent แล้ว closed itemsets ขนาด k+1 ที่เป็น Superset ของ closed itemsets นั้นจะไม่เป็น frequent ด้วย" หรือกล่าวอีกนัยหนึ่งว่า " ถ้า closed itemset ที่เป็น Superset เป็น frequent แล้ว closed itemsets ที่เป็นสับเซตจะเป็น frequent ด้วย " และ "เซตของ frequent itemsets ที่ยาวที่สุดจะเท่ากับเซตของ frequent closed itemsets ที่ยาวที่สุด" โดยการเลือกใช้โครงสร้างข้อมูลที่เหมาะสมเพื่อลดการใช้ memory ในการจัดเก็บข้อมูลให้น้อยลงนั่นเอง

2.4.1 คำนิยามและหลักการที่เกี่ยวข้องแบบ Lattice

2.4.1.1 นิยามเทอม

กำหนดนิยามเพื่ออธิบายกฎความสัมพันธ์ดังนี้ ให้ $I = \{i_1, i_2, \ldots, i_n\}$ เป็นเซตของ Items และ D เป็นเซตของ Transactions โดยที่ Transaction T เป็นสับเซตของ I $(T \subseteq I)$ มี *TID* เป็นหมายเลขประจำ Transaction คำว่า Itemset หมายถึง เซตของ Items และ k-Itemset หมายถึง เซตของ Items ที่มีสมาชิก k ตัว ค่าความถี่ของ Itemset X หมายถึงจำนวน Transactions ที่มี X ปรากฏอยู่ ถ้าความถี่ของ Itemset มีค่ามากกว่าหรือเท่ากับค่าความถี่ขั้นต่ำจะเรียก Itemset นั้น ๆ ว่า Frequent Itemset กฎ

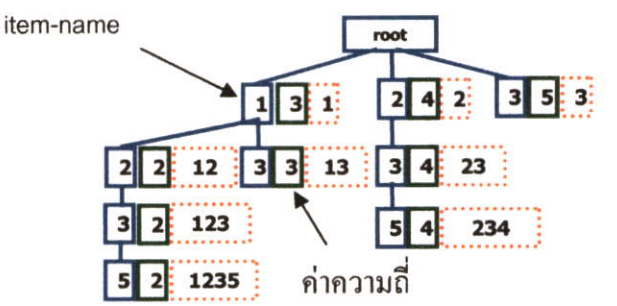
ความสัมพันธ์แสดงได้ในรูปแบบ $X \Rightarrow Y$ โดยที่ $X \subset I$, $Y \subset I$ และ $X \cap Y = \varnothing$ X และ Y เป็น Itemsets ใด ๆ และกฎความสัมพันธ์ที่ยอมรับได้ จะต้องมีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ

2.4.1.2 Closed itemset

ให้ itemset Y เป็นสับเซตของ *I* ($Y \subseteq I$) และ itemset Y จะเป็น Closed itemset ก็ต่อเมื่อไม่มี itemset Y' ที่เป็นซุปเปอร์เซตของ itemset Y และค่าความถี่ของ itemset Y' เท่ากับ itemset Y และถ้า itemset Y จะเป็น frequent itemset ก็ต่อเมื่อค่าความถี่ของ itemset Y มากกว่าหรือเท่ากับค่าความถี่ขั้นต่ำ

2.4.1.3 Frequent closed itemset tree (FC Tree)

จากรูปที่ 2.4 แสดงถึง FC Tree ซึ่งเป็น prefix tree ที่ใช้เก็บ frequent closed itemsets โดยข้อมูลที่เก็บในแต่ละโหนดคือ item-name และค่าความถี่ของ path นั้น ๆ ตัวอย่างเช่น path ซ้ายสุดจะได้ frequent closed itemset {1,2,3,5} มีความถี่ 2 ส่วน สี่เหลี่ยมจุดไข่ปลานั้นเป็นเพียงการแสดง itemset เพื่อง่ายต่อการอธิบาย ไม่ได้มีการเก็บไว้จริงในโครงสร้างข้อมูล



รูปที่ 2.4 โครงสร้างข้อมูล FC Tree

2.4.2 โครงสร้างข้อมูลและอัลกอริทึมแบบ Lattice

ในส่วนนี้จะอธิบายถึง FC Tree ซึ่งใช้ในและจัดเก็บ closed itemsets และจะอธิบายอัลกอริทึม Close พร้อมทั้งยกตัวอย่างโดยใช้ตัวอย่าง ข้อมูลขนาดเล็ก เพื่อให้ง่ายต่อการทำความเข้าใจ

2.4.2.1 โครงสร้างข้อมูลแบบ Lattice ประกอบด้วย 2 ส่วน ดังนี้

2.4.2.1.1 FCC_i (Frequent candidate closed itemsets) ประกอบด้วย itemsets ขนาด i, candidate closed itemsets ของ itemsets ซึ่งในที่นี้จะเรียกว่า closure และ ค่าความถี่ของ closed itemsets

2.4.2.1.2 FC (Frequent closed itemsets) ในส่วนนี้จะเป็นโครงสร้างต้นไม้ ซึ่งประกอบด้วย frequent closed itemsets ทั้งหมด โดยที่แต่ละโหนดประกอบด้วย item-name และค่าความถี่ของ frequent closed itemsets

2.4.2.2 อัลกอริทึม แบบ Lattice ประกอบด้วย 3 ขั้นตอนใหญ่ โดยขั้นตอนที่ 1 และ 2 ใช้การทำซ้ำ คือ

2.4.2.2.1 การสร้าง itemsets ขนาด i เพื่อที่จะนำไปหา closed itemsets ใน FCC ลำดับที่ i โดยหลักการในการสร้าง คือ ลำดับที่ 1 (i=1) สร้างจาก frequent itemsets ขนาด 1 และลำดับที่ 2 (i=2) สร้างได้จากการนำ frequent itemset ลำดับที่ 1 มารวมกัน และลำดับที่ 3 ขึ้นไป (i>=3) สร้างได้จากลำดับที่ i-1 มารวมกัน โดยที่ itemsets ที่จะรวมกันได้ต้องมีสมาชิกตัวแรกถึงตัวรองสุดท้ายเหมือนกัน (ABC + ABE = ABCE) และ itemsets ที่ได้จากการรวมนั้น ต้องผ่านการตรวจสอบ 2 ขั้นตอน คือ (1) ซับเซตขนาด i-1 ทุกตัวของ itemset นั้น ต้องเป็นสมาชิกของ itemsets ใน FCC ลำดับที่ i-1 (2) itemset นั้น ต้องไม่เป็นซับเซตของ frequent closed itemsets ของซับเซตขนาด i-1 ของ itemset นั้น โดยการทำงานของอัลกอริทึมจะสิ้นสุดเมื่อไม่สามารถหา itemset ขนาด i ได้ใน FCC ลำดับที่ i ใด ๆ

2.4.2.2.2 การหา closed itemsets และค่าความถี่ของ closed itemsets จาก itemtets ใน FCC ลำดับที่ i ที่สร้างได้จากขั้นตอนที่ 1 และในขั้นตอนนี้จะทำการตัด itemsets และ Closed itemsets ที่มีค่าความถี่ต่ำกว่าค่าความถี่ขั้นต่ำ จากนั้นจะนำ frequent closed itemsets ที่ได้มาสร้าง FC (Frequent closed itemsets) ซึ่งเป็น Prefix Tree

Tid	Items
1	1 3 4
2	2 3 5
3	1 2 3 5
4	2 3 5
5	1 2 3 5

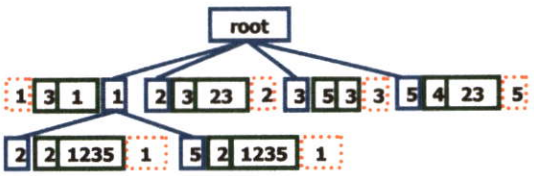
(1) ตัวอย่างฐานข้อมูล

Itemset	Closure	Support
{1}	{1, 3}	3
{2}	{2, 3, 5}	4
{3}	{3}	5
{5}	{2, 3, 5}	4

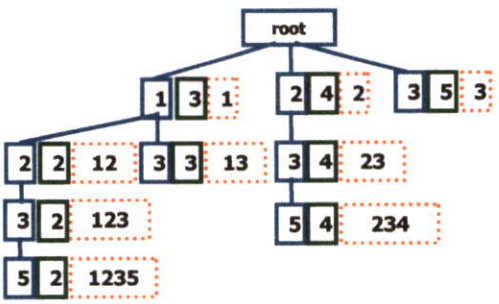
(2) FCC ลำดับที่ 1

Itemset	Closure	Support
{1, 2}	{1, 2, 3, 5}	2
{1, 5}	{1, 2, 3, 5}	2

(3) FCC ลำดับที่ 2



(4) โครงสร้าง FC Tree แบบเก่า



(5) โครงสร้าง FC Tree แบบใหม่

รูปที่ **2.5** ตัวอย่างฐานข้อมูลและ FCC แต่ละลำดับ

2.4.2.2.3 การ Traverse FC เพื่อจำแนก frequent closed itemsets ทั้งหมดตามขนาดและการหาซับเซตทุกขนาดของ frequent closed itemsets เพื่อให้ได้ frequent itemsets และค่าความถี่ที่ถูกต้องและครบถ้วน

ตัวอย่าง จากข้อมูลในตาราง รูปที่ 2.5 (1) กำหนดค่าความถี่ขั้นต่ำ คือ 2 Transactions เริ่มต้นด้วยการหา frequent itemsets ขนาด 1 เพื่อเป็น itemsets ใน FCC ลำดับที่ 1 หลังจากนั้นเข้าสู่ลูปในการทำซ้ำ โดยเริ่มจากขั้นตอนที่ 3 คือการหา closure ของ itemsets ใน FCC ลำดับที่ 1 โดยใช้การ intersection itemsets ที่เป็นสมาชิกของแต่ละ transaction โดย transaction ที่จะสามารถ intersection กันได้ต้องเป็นซุปเปอร์เซตของ itemsets ใน FCC ลำดับที่ 1 เช่น พิจารณา {1} ใน FCC ลำดับที่ 1 กับ itemsets ใน transaction ที่ 1 คือ {1,3,4} ซึ่งเป็นซุปเปอร์เซตของ {1} และ closure ของ {1} เป็น NULL จึงได้ผลลัพธ์คือ {1} มี closure คือ {1,3,4} ค่าความถี่เท่ากับ 1 และพิจารณาเช่นเดียวกันจนครบทุก itemsets ใน FCC ลำดับที่ 1 และจึงเริ่มต้นพิจารณา {1} กับ transaction ต่อไปเช่นเดิมอีก ในที่นี้จะขอยกตัวอย่างแค่การหา closure ของ {1} พิจารณา {1} หลังจากผ่าน transaction ที่ 1 แล้ว {1} มี closure คือ {1,3,4} ค่าความถี่เท่ากับ 1 กับ itemsets ใน transaction ที่ 2 คือ {2,3,5} แต่ไม่เป็นซุปเปอร์เซตของ {1} จึงไม่ต้อง intersection กัน พิจารณา itemsets ใน transaction ที่ 3 คือ {1,2,3,5} ซึ่งเป็นซุปเปอร์เซตของ {1} ทำการ intersection closure ของ {1} กับ itemsets ใน transaction ที่ 3 ({1,3,4} $\cap$ {1,2,3,5}) จะได้ {1} มี closure คือ {1,3} ค่าความถี่เท่ากับ 2 ต่อไปพิจารณา itemsets ใน transaction ที่ 4 แต่ไม่เป็นซุปเปอร์เซตของ {1} จึงพิจารณา itemsets ใน transaction ที่ 5 คือ {1,2,3,5} เป็นซุปเปอร์เซตของ {1} ทำการ intersection ({1,3} $\cap$ {1,2,3,5}) ได้ {1} มี closure คือ {1,3} ค่าความถี่เท่ากับ 3 เป็นผลลัพธ์ และเมื่อพิจารณาเช่นเดียวกันครบทุก itemsets ใน FCC ลำดับที่ 1 แล้วจะได้ FCC ลำดับที่ 1 ดังรูปที่ 2.5 (2) หลังจากนั้นทำการตัด itemsets และ closure ที่มีค่าความถี่ต่ำกว่าค่าความถี่ขั้นต่ำคือ 2 แต่พิจารณาแล้วค่าความถี่ของ closure ใน FCC ลำดับที่ 1 ผ่านค่าความถี่ขั้นต่ำทุกตัว จึงทำการสร้าง FC tree ตาม FCC ลำดับที่ 1 หลังจากนั้นทำการสร้าง itemsets ขนาด 2 เพื่อเป็น itemsets ใน FCC ลำดับที่ 2 จาก

itemsets ใน FCC ลำดับที่ 1 โดยใช้หลักการรวมกันและต้องผ่านการตรวจสอบ 3 ข้อ ที่ได้กล่าวไปในหัวข้อ 2.4.2.1 ข้อ 1 โดยการรวมจะได้ itemsets ดังนี้ {1,2}, {1,3}, {1,5}, {2,3}, {2,5}, {3,5} แต่เนื่องจาก {1,3} เป็นซับเซตของ closure ของ {1} และ {2,3}, {2,5}, {35,} เป็นซับเซตของ closure ของ {2} และ {5} จึงตัดทิ้งไม่นำไปเป็น itemsets ใน FCC ลำดับที่ 2 ดังนั้นจะได้ itemsets ขนาด 2 ใน FCC ลำดับที่ 2 คือ {1,2}, {1,5} หลังจากนั้นทำการหา closure เช่นเดิม เมื่อทำการพิจารณาครบทุก transaction แล้วจะได้ผลลัพธ์ดังรูปที่ 2.5 (3) เพื่อที่จะนำไปสร้าง FC Tree และต่อไปทำการสร้าง itemsets ขนาด 3 เพื่อที่จะเป็น itemsets ใน FCC ลำดับที่ 3 จากการรวมกันของ itemsets ใน FCC ลำดับที่ 2 จะได้ {1,2,5} แต่เนื่องจาก {1,2,5} เป็นซับเซตของ closure ของ {1,2}, {1,5} ดังนั้นจึงตัดทิ้ง ทำให้ไม่มี itemsets ใน FCC ลำดับที่ 3 จึงเป็นการสิ้นสุดการวนลูป หลังจากสิ้นสุดการวนลูปแล้วจะได้ FC Tree ดังรูปที่ 2.5 (4) ซึ่งมีการจัดเก็บข้อมูลเกินความจำเป็น จึงได้นำเสนอ FC Tree แบบใหม่ดังรูปที่ 2.5 (5) ซึ่งจัดเก็บแต่ closure และค่าความถี่ของ closure ซึ่งช่วยลดหน่วยความจำในการเก็บ itemsets ของ closure ลงได้ และในขั้นตอนสุดท้ายเป็นการ traverse FC Tree เพื่อแยก frequent closed itemsets ที่ได้ออกตามขนาดและทำการหาซับเซตขนาดรองลงมา เช่น เส้นทางที่ยาวที่สุดคือ {1, 2, 3, 5}จะได้ frequent itemsets ขนาด 4 คือ {1,2,3,4} ความถี่ 2 และแยกซับเซตได้ {1,2,3} ความถี่ 2 {1,2,5} ความถี่ 2 {1,3,5} ความถี่ 2 {2,3,5} ความถี่ 4 พิจารณาที่ {2,3,5} จากการแยกซับเซตควรมีค่าความถี่ 2 แต่ถ้าพิจารณาจาก FC Tree แล้วจะให้ว่ามี path {2,3,5} ซึ่งมีความถี่ 4 ซึ่งมากกว่า 2 จึงได้ {2,3,5} มีความถี่ 4 พิจารณาและแยกซับเซตไปจนครบก็จะได้ frequent itemsets ครบทุกขนาด

2.5 การค้นหาข้อมูลแบบลิเนียร์เซิร์ท (Linear search)

เมื่อให้ข้อมูล (Data) ประกอบด้วยส่วนประกอบต่าง ๆ แต่ละส่วน(Elements) ที่เก็บอยู่ในหน่วยความ จำและกลุ่มข้อมูล (Item) เป็นข้อมูลที่กำหนดให้การค้นหาเป็นการหาตำแหน่งของกลุ่มข้อมูลที่ อยู่ในหน่วยความจำนั้น หรือแสดงข้อความให้ทราบหากกลุ่มของข้อมูลไม่อยู่ในหน่วยความจำนั้น ๆ การค้นหาจะประสบความสำเร็จเมื่อมีข้อมูลอยู่ในหน่วยความจำและจะถือว่าไม่ประสบความสำเร็จเมื่อเป็นไปตามกรณีอื่น ๆ

บ่อยครั้งที่อาจมีการเพิ่มรายการในกลุ่มข้อมูลลงในหน่วยความจำ หากไม่พบกลุ่มข้อมูลหลังจากการค้นหา จึงส่งผลทำให้มีการใช้การค้นหาและการแทรกข้อมูลร่วมกัน แทนที่จะใช้การค้นหาเพียงอย่างเดียวเท่านั้น

ในความเป็นจริงแล้วมีอัลกอริทึมที่ใช้สำหรับการค้นหาข้อมูลอยู่หลายแบบด้วยกัน ซึ่งการเลือกใช้จะเป็นแบบใดนั้นขึ้นอยู่กับโครงสร้างของการเก็บข้อมูลเป็นหลัก แต่สำหรับในวิทยานิพนธ์ฉบับนี้จะขอกล่าวถึงอัลกอริทึมที่ง่ายที่สุด คือ การค้นหาแบบลิเนียร์เซิร์ท นั่นเอง

สำหรับการค้นหาแบบลิเนียร์เซิร์ทนั้นจะเริ่มต้นจากการกำหนดข้อมูลที่มีลักษณะเป็นอาร์เรย์แบบต่อเนื่อง ซึ่งมีขนาดเท่ากับ n อิลิเมนท์ การค้นหาข้อมูลในแต่ละกลุ่มของข้อมูลหรือในแต่ละชุดข้อมูลจากข้อมูลหลักที่ได้ทำการเก็บเอาไว้ในส่วนของหน่วยความจำ สามารถทำได้โดยการเปรียบเทียบกับกลุ่มของข้อมูลย่อยแต่ละชุดกับแต่ละอิลิเมนท์ ของข้อมูลหลักนั่นก็คือ จะทำการเปรียบเทียบข้อมูลไปเรื่อยจนกว่าจะพบข้อมูลของตัวที่ต้องการซึ่งวิธีนี้จะต้องเข้าถึงข้อมูลในทุก ๆ อิลิเมนท์ของข้อมูลที่อยู่ในหน่วยความจำโดยวิธีการแบบนี้เรียกว่าเป็นวิธีการค้นหาแบบลิเนียร์เซิร์ท (Linear search)

ความซับซ้อนของอัลกอริทึมแบบลิเนียร์เซิร์ทจะวัดจากค่าของ f(n) ที่สามารถหาได้ จากการเปรียบเทียบกลุ่มหรือชุดของข้อมูลกับข้อมูลหลัก เมื่อข้อมูลหลักนั้นมีจำนวน n อิลิเมนท์ ดังนั้นจะมีกรณีที่สามารถเกิดขึ้นได้ 2 กรณี คือ กรณีที่แย่ที่สุด (Worst case) และกรณีที่เกิดขึ้นทั่ว ๆ ไป (Average case) สำหรับในกรณีที่แย่ที่สุดจะปรากฏเมื่อไม่มีชุดหรือกลุ่มของข้อมูลอยู่ในหน่วยความจำซึ่งการเปรียบเทียบที่เกิดขึ้นก็จะทำการเปรียบเทียบกับข้อมูลทั้งหมด ดังนั้น

$$f(n) = n + 1 \qquad (2.1)$$

เพราะฉะนั้นเวลาที่ใช้ในการทำงานจะประมาณ n ส่วนในกรณีทั่ว ๆ ไป (Average case) จะใช้หลักการของความน่าจะเป็นดังนี้ สมมติให้ P_k เป็นความน่าจะเป็นที่กลุ่มของข้อมูลจะอยู่ในหน่วยความจำ Data[K] และ q เป็นความน่าจะเป็นที่กลุ่มข้อมูลไม่ปรากฏในหน่วยความจำ เพราะฉะนั้น จะได้

$$p_1 + p_2 + \cdots q = 1 \qquad (2.2)$$

เนื่องจากอัลกอริทึมใช้การเปรียบเทียบเป็นจำนวน K ครั้ง หากมีกลุ่มข้อมูลในหน่วยความจำ (Data[K]) ค่าเฉลี่ยของการเปรียบเทียบจะได้จาก

$$f(n) = 1 \cdot p_1 + 2 \cdot p_2 + \cdots + n \cdot p_n + (n+1) \cdot q \qquad (2.3)$$

สมมติให้ q น้อยมากและกลุ่มข้อมูลที่ปรากฏในแต่ละอิลิเมนท์มีความน่าจะเป็นที่เท่ากัน ดังนั้นจึงส่งผลทำให้ $q \approx 0$ และแต่ละ $p_i = 1/n$ ดังนั้น

$$f(n) = 1 \cdot \frac{1}{n} + 2 \cdot \frac{1}{n} + \cdots + n \cdot \frac{1}{n} + (n+1) \cdot 0 = (1 + 2 + \cdots + n) \cdot \frac{1}{n}$$

$$= \frac{n(n+1)}{2} \cdot \frac{1}{n} = \frac{n+1}{2} \qquad (2.4)$$

เพราะฉะนั้นค่าเฉลี่ยของการเปรียบเทียบในอัลกอริทึมจะมีค่าประมาณครึ่งหนึ่ง ของจำนวนอิลิ-เมนท์ ในอาร์เรย์และเพื่อให้การทำงานง่ายขึ้นในตอนต้นจะทำการกำหนดกลุ่มข้อมูลให้ปรากฏอยู่ในส่วนที่ เป็นหน่วยความจำ ณ ตำแหน่ง N+1 (Data [N+1]) ซึ่งเป็นตำแหน่งต่อจากท้ายสุดของ Data ซึ่งจะได้ LOC = N+1เมื่อ LOC คือตำแหน่งของ Item ที่ปรากฏอยู่ใน Data ตัวแรก เพื่อเป็นการแสดงว่า ไม่มีข้อมูล Item อยู่ใน Data สำหรับจุดประสงค์ของการกำหนดในลักษณะนี้ เพื่อป้องกันการตรวจสอบซ้ำเมื่อมาถึงข้อมูลท้ายสุด ซึ่งวิธีการนี้จะพบข้อมูลทุกครั้ง [13]

2.6 การค้นหาข้อมูลไบนารีทรี (Binary trees)

เมื่อพูดถึงต้นไม้ (Tree) แล้ว ทุกคนคงจะต้องนึกถึงสิ่งมีชีวิตประเภทพืช ซึ่งมีส่วนประกอบเหมือนกันคือราก ลำต้น ใบ และกิ่ง ไม่ว่าต้นไม้นั้น ๆ จะมีขนาดเล็กหรือใหญ่ก็ตาม

สำหรับโครงสร้างข้อมูลแบบต้นไม้ในคอมพิวเตอร์ที่กล่าวถึงนี้ จะหมายถึงโครงสร้างข้อมูลชนิดหนึ่งที่มีการเก็บข้อมูลต่าง ๆ เอาไว้ในโหนด (Node) โดยโครงสร้างข้อมูลแบบต้นไม้นี้จะมีการแบ่งออกเป็น 2 ประเภท คือ รูทโหนด (Root node) และโหนดลูก (Child node) โดยที่ต้นไม้ 1 ต้น จะประกอบไปด้วยรูทโหนด เพียงโหนดเดียวเท่านั้น ส่วนโหนดที่เหลือจะจัดเป็นโหนดลูกทั้งหมด ซึ่งสามารถมีได้ตั้งแต่ 0 โหนดขึ้นไป

ถ้าลองทำการเปรียบเทียบโครงสร้างข้อมูลแบบต้นไม้กับโครงสร้างข้อมูล แบบสแตก (Stack) คิว (Queue) และลิงค์ลิสต์ (Linked list) แล้วนั้น จะพบว่าการจัดเก็บและการค้นหาข้อมูลของโครงสร้างข้อมูลแบบต้นไม้จะทำงานได้ค่อนข้างเร็วกว่า โครงสร้างของข้อมูลชนิดอื่น ๆ ที่ ได้กล่าวมาข้างต้น เนื่องจากโครงสร้างของข้อมูลแบบต้นไม้นั้น ใช้หลักการเข้าถึงข้อมูลแบบ ไบนารีเซิร์ททรี (Binary search trees) นั่นเอง ในขณะที่โครงสร้างข้อมูลอีก 3 วิธีการที่กล่าวมาแล้วนั้นจะใช้หลักการเข้าถึงข้อมูลแบบตามลำดับ (Sequential search) [14] เป็นต้น

ไบนารีทรี ให้ T เป็นเซตของอิลิเมนท์ต่าง ๆ ที่เรียกว่าโหนด ซึ่งมี 2 ลักษณะคือ

1) T ว่างไม่มีข้อมูล เรียกว่า Null tree หรือ Empty tree

2) T ประกอบด้วยโหนด R เรียกว่า รากของต้นไม้ (Root of T) และโหนดที่เหลือของ T จะสร้างคู่ทรีที่แยกย่อยออกไปเป็นไบนารี T1 และ T2

ถ้า T ประกอบด้วยราก R จะมี T1 และ T2 ซึ่งเรียกว่า ทรีย่อยซ้าย และ ทรีย่อยขวา (Left and right subtrees) ของ R หาก T1 มีข้อมูลบรรจุอยู่จะเรียกว่า ซัคเซสเซอร์ซ้าย (Left successor) ของ R และหาก T2 มีข้อมูลอยู่จะเรียกว่า ซัคเซสเซอร์ขวา (Right successor) ของ R

ไบนารีทรี สามารถแสดงได้ดังไดอะแกรมในรูปที่ 2.6 โครงสร้างของไบนารีทรี T จะเป็นดังนี้ (i) T ประกอบด้วย 11 โหนด ในแต่ละโหนดแสดงด้วยตัวอักษร A-L ยกเว้น I (ii) รากของ T คือ โหนด A ซึ่งอยู่บนสุดเส้นลงเอียงไปทางซ้ายจากโหนด N ใด ๆ แสดงถึงซัคเซสเซอร์ซ้าย ของ N และเส้นลงเอียงไปทางขวาจากโหนด N ใด ๆ เป็นซัคเซสเซอร์ขวา ของ N ดังนั้น

1. B เป็นซัคเซสเซอร์ซ้าย และ C เป็นซัคเซสเซอร์ขวา ของโหนด A

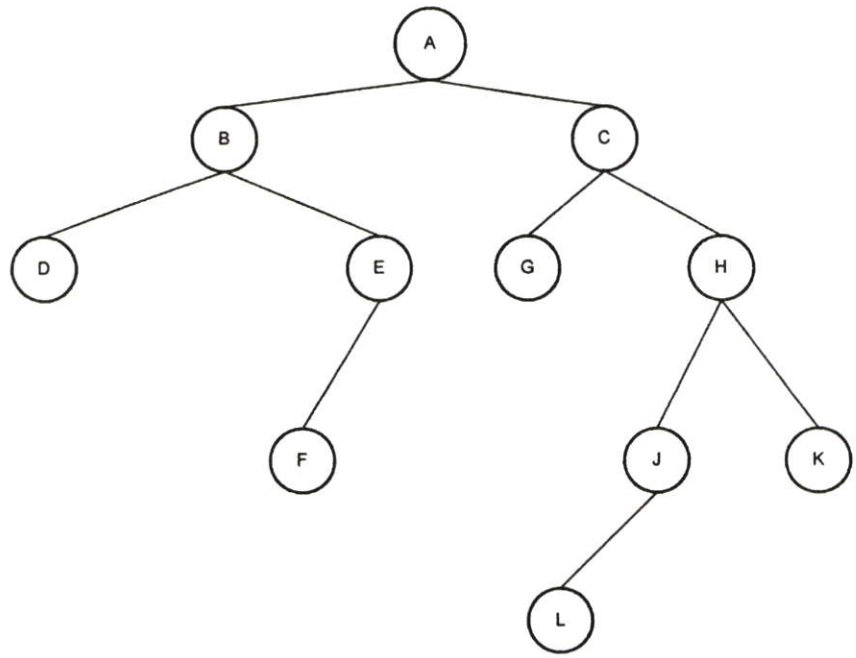
2. ทรีย่อยด้านซ้ายของราก A ประกอบด้วย โหนด B, D, E และ F และทรีย่อยด้านขวา ของ A ประกอบด้วยโหนด C, G, H, J, K และ L

โดยในแต่ละโหนด N ในไบนารีจะมี 0, 1 หรือ 2 ซัคเซสเซอร์ก็ได้ โหนด A, B, C และ H มี 2 ซัคเซสเซอร์โหนด E และ J จะมีเพียงซัคเซสเซอร์เดียว และโหนด D, F, G, L และ K ไม่มีซัคเซสเซอร์สำหรับโหนดที่ไม่มีซัคเซสเซอร์ จะเรียกว่าโหนดท้าย (Terminal node)

จากคำจำกัดความที่ได้กล่าวมาทั้งหมดของไบนารีทรี T จะเป็นรีเคอร์ซีฟ เนื่องจาก T ถูกกำหนดอยู่ในเทอมของทรีย่อย T1 และ T2 นั่นคือ ทุก ๆ โหนด N ของ T จะมีทรีย่อยด้านซ้าย และขวา หาก N เป็นโหนดท้ายสุด ทรีย่อยซ้าย และ ขวาก็จะว่าง

ในการเขียนโปรแกรมจัดการกับไบนารีทรี โดยการนำเอาอาร์เรย์ เข้ามาช่วย จำเป็นจะต้องมีการจองขนาดของอาร์เรย์ไว้ก่อนล่วงหน้า ซึ่งตามปกติการจองขนาดของอาร์เรย์ก็ทำเพื่อรองรับจำนวนข้อมูลให้มีประสิทธิภาพมากที่สุดและการคำนวณขนาดของอาร์เรย์ที่เหมาะสม จะมีการนำเอาหลักการของไบนารีทรีแบบสมบูรณ์ (Complete binary tree) เข้ามาช่วย ดังจะได้กล่าวถึงรายละเอียดต่อไป

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

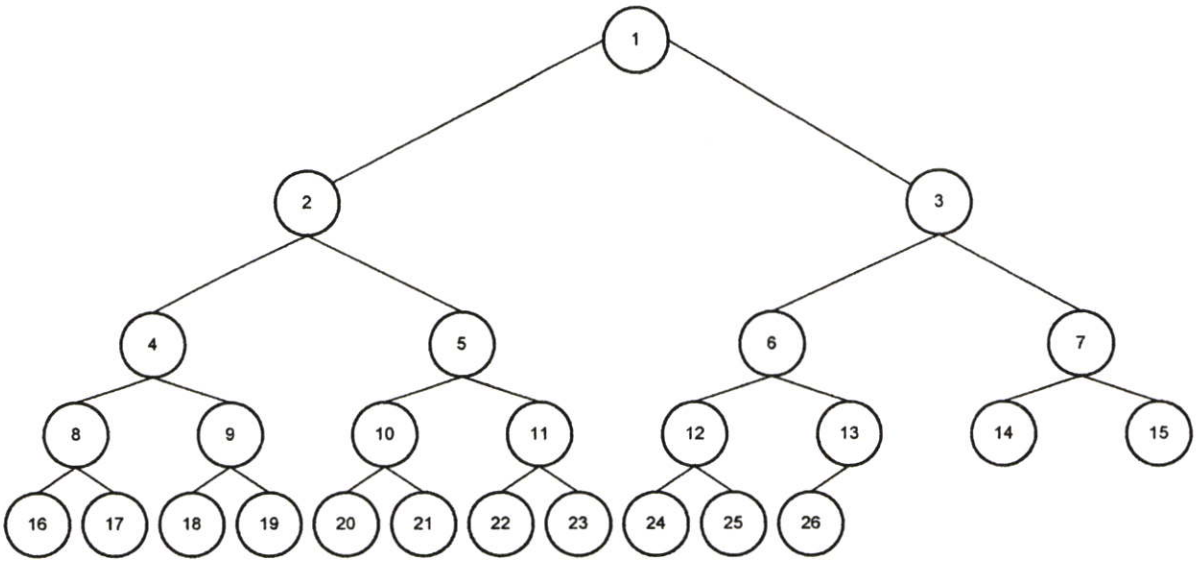


รูปที่ 2.6 โครงสร้างของไบนารีทรี

2.7 ไบนารีทรีแบบสมบูรณ์ (Complete binary tree)

กล่าวคือไบนารีทรี T ใด ๆ ในแต่ละโหนดของ T จะมีโหนดลูกสูงสุดด้วยกัน 2 โหนด โดยสามารถคำนวณหาจำนวนโหนดได้จาก 2^r เมื่อ r เป็นหมายเลขระดับของโหนด ซึ่งจะเรียกว่าเป็น T ทรีสมบูรณ์ก็ต่อเมื่อในแต่ละระดับมีโหนดสูงสุด ยกเว้นโหนดสุดท้าย รูปที่ 2.7 แสดงทรีที่สมบูรณ์ของ T_{26} [14] ซึ่งมี 26 โหนด (ไม่พิจารณาข้อมูลในโหนด)

74826



รูปที่ 2.7 ทรีที่สมบูรณ์ของ T_{26}

โหนดของทรี T_{26} ในรูปที่ 2.7 สามารถเขียนด้วยตัวเลขกำกับตั้งแต่ 1, 2, ... , 26 จากซ้ายไปขวา โดยการกำหนดหมายเลขกำกับแบบนี้ สามารถทำให้มองเห็นโหนดลูกและโหนดแม่ใน T_n ได้ง่าย โดยโหนดลูกด้านซ้ายและขวาของโหนด K กำหนดจาก 2 * K และ 2 * K+1 และโหนดแม่ของ K คือโหนด [K/2] ตัวอย่างเช่น โหนดลูกของโหนด 9 คือ โหนด 18 และ 19 และโหนดแม่คือโหนด [9/2] = 4 ดังนั้นความลึกของทรี T_n ที่มี n โหนด สามารถกำหนดได้จากสมการ

$$D_n = \lfloor \log_2 n + 1 \rfloor \qquad (2.5)$$

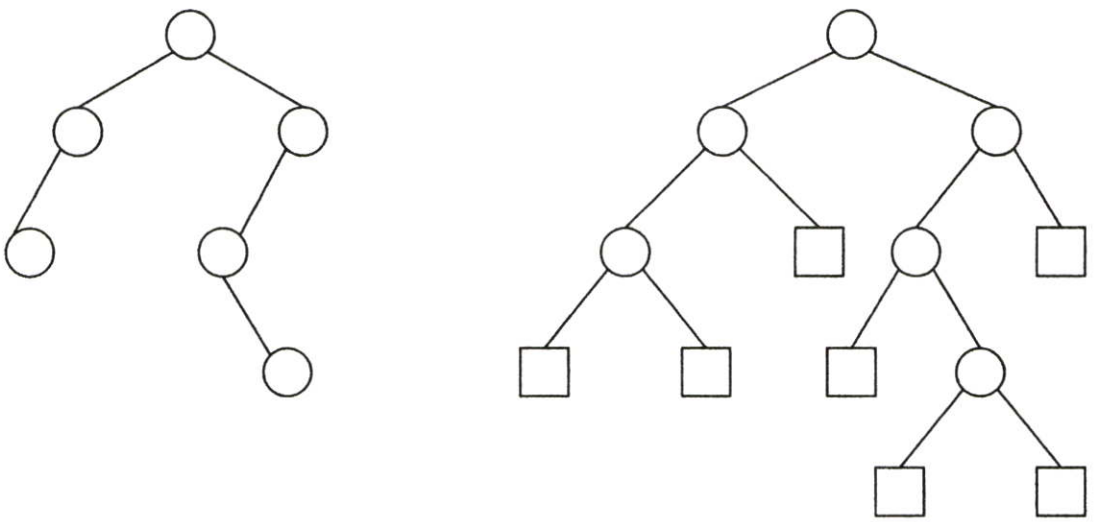
เช่น T_n มี $n = 1000000$ โหนด จะมีความลึก $D_n = 21$

2.8 ไบนารีทรีแบบขยาย : 2-ทรี (Extended binary trees : 2 – trees)

ไบนารีทรี T เรียกว่าเป็น 2-ทรี เมื่อแต่ละโหนด N มีโหนดลูก 0 หรือ 2 โหนด ในกรณีที่มีโหนดลูก 2 โหนด จะเรียกว่าโหนดใน (Internal node) และโหนดที่มีโหนดลูกเป็น 0 จะเรียกว่าโหนดนอก (External node) ในบางกรณีอาจจะใช้วงกลมแทนโหนด n และสี่เหลี่ยมแทนโหนดนอก

เทอมของไบนารีทรีแบบขยายมาจากการทำงานดังต่อไปนี้ สมมติให้ไบนารีทรี T ดังแสดงในรูป ที่ 2.8 (a) โดยทรี T สามารถเปลี่ยนไปเป็น 2-ทรีได้ โดยการแทนแต่ละทรีย่อยที่ว่างด้วย

โหนดใหม่ ดังแสดงในรูปที่ 2.8 (b) สังเกตโหนดใหม่ที่เพิ่มขึ้นมาเป็นโหนดนอกและโหนดเก่าจะเป็นโหนดใน



(a) Binary tree T (b) Extended 2-Tree

รูปที่ 2.8 การแปลงไบนารีทรี T เป็น 2-ทรี

ในลำดับต่อไปจะขอกล่าวถึงในส่วนของการเก็บไบนารีทรีในหน่วยความจำ ซึ่งในหัวข้อนี้จะเป็นการแสดงการเก็บไบนารีทรีในหน่วยความจำ 2 แบบด้วยกัน คือ แบบแรกใช้ลักษณะการเก็บแบบลิงค์ลิสต์ และแบบที่ 2 จะใช้ลักษณะการเก็บแบบอาร์เรย์ ซึ่งเป็นการเก็บแบบเรียงตามลำดับ โดยการเก็บทรีในหน่วยความจำจะเป็นแบบใดนั้น ก็จะขึ้นอยู่กับความต้องการการเข้าถึงราก R ของ T ได้โดยตรง และเมื่อกำหนดโหนด N ของทรี T ให้ จะสามารถเข้าถึงโหนดลูกของ N ได้โดยตรง ซึ่งในที่นี้จะขอกล่าวถึงรูปแบบในการเก็บข้อมูลแบบลิงค์ลิสต์เท่านั้น

2.9 การเก็บไบนารีทรีแบบลิงค์ลิสต์

หากไม่มีการกำหนดข้อมูลเป็นอย่างอื่นใดอัลกอริทึมของไบนารีทรีนี้จะเก็บข้อมูลในลักษณะของลิงค์ ซึ่งใช้อาร์เรย์ 3 อาร์เรย์ขนานกัน คือ INFO, LEFT และ RIGHT และใช้ตัวแปรที่เป็นพ้อยน์เตอร์ (Pointer) ROOT แต่ละโหนด N ของ T จะสัมพันธ์กับตำแหน่งของอาร์เรย์ K ดังต่อไปนี้

2.9.1 INFO [K] เก็บข้อมูลของโหนด N

2.9.2 LEFT [K] เก็บตำแหน่งของโหนดลูกทางด้านซ้ายของโหนด N

2.9.3 RIGHT [K] เก็บตำแหน่งของโหนดลูกทางด้านขวาของโหนด N

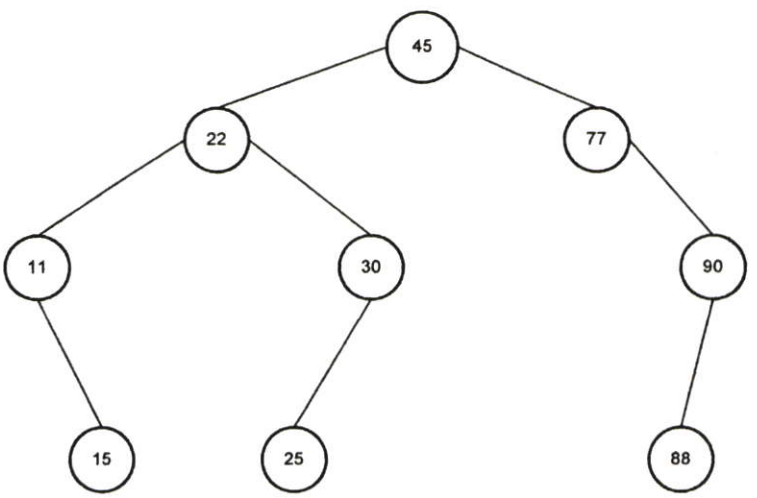
ส่วนตัวแปร ROOT จะเก็บตำแหน่งของราก R ของ T ถ้าทรีย่อยว่าง ตัวชี้ตำแหน่งจะเก็บค่า NULL ดังนั้นถ้าทรี T ว่าง จะส่งผลทำให้ค่า ROOT จะเป็น NULL นั่นเอง

หมายเหตุ 1: ข้อมูลในแต่ละโหนดจะมีเพียงรายการเดียวเท่านั้น ซึ่งในการใช้งานจริง ข้อมูลในโหนดอาจจะเป็นเรคคอร์ดที่เก็บอยู่ใน INFO ก็ได้

หมายเหตุ 2 : เนื่องจากโหนดอาจจะมีการแทรกและลบออกจากทรี สามารถกำหนดตำแหน่งว่างในอาร์เรย์ INFO, LEFT และ RIGHT ในลิงค์ลิสท์ด้วยพ้อยน์เตอร์ AVAIL (AVAIL คือ pointer ที่ชี้ไปยังตำแหน่งที่ต้องการ) และใช้อาร์เรย์: LEFT เป็นตัวพ้อยน์เตอร์ของ AVAIL

หมายเหตุ 3: ใช้ค่า 0 หรือค่าลบในการแสดงค่า NULL

นอกจากการเก็บข้อมูลแบบลิงค์ลิสท์แล้ว สิ่งที่ต้องรู้และเข้าใจในลำดับต่อไปก็คือการเข้าถึงข้อมูลในไบนารีทรี (Traversing binary trees) การเข้าถึงข้อมูลในไบนารีทรี T ที่มีราก R มีอยู่ด้วยกัน 3 แบบ คือ พรีออร์เดอร์ อินออร์เดอร์ และโพสท์ออร์เดอร์ (Preorder, Inorder and Postorder) [15] ตามลำดับ



รูปที่ 2.9 ลำดับการเข้าถึงข้อมูล

สำหรับลำดับการเข้าถึงข้อมูลในแบบต่าง ๆ สามารถแสดงได้ดังรูปที่ 2.9 และสามารถนำเอามาเขียนย่อเป็นรูปแบบที่ง่ายต่อการเข้าใจได้ดังนี้

พรีออร์เดอร์ : (1) ประมวลผลจากราก R
(2) ค้นหาทรีย่อยทางขวาซ้ายของ R ในพรีออร์เดอร์
(3) ค้นหาทรีย่อยทางขวาของ R ในพรีออร์เดอร์

อินออร์เดอร์ : (1) ค้นหาทรีย่อยทางซ้ายของ R ในอินออร์เดอร์
(2) ประมวลผลจากราก R
(3) ค้นหาทรีย่อยทางขวาของ R ในอินออร์เดอร์

โพสท์ออร์เดอร์ : (1) ค้นหาทรีย่อยทางซ้ายของ R ในโพสท์ออร์เดอร์

(2) ค้นหาทรีย่อยทางขวาของ R ในโพสท์ออร์เดอร์

(3) ประมวลผลจากราก R

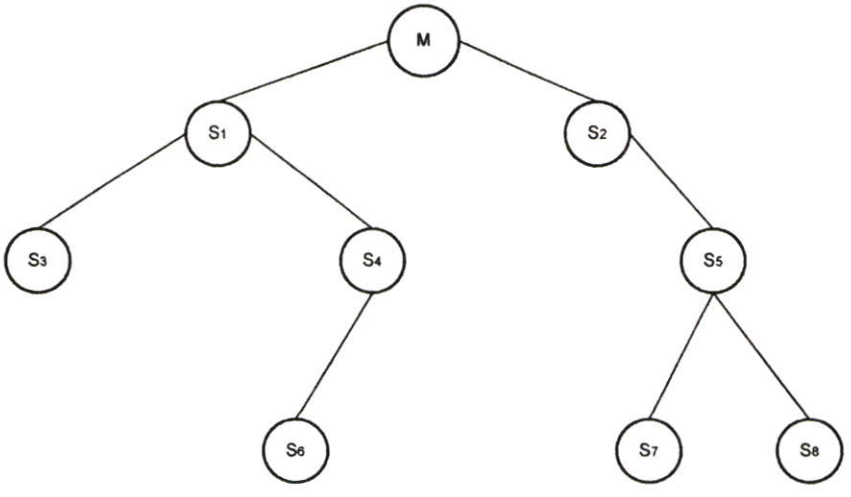
สังเกตว่าในแต่ละอัลกอริทึมจะประกอบไปด้วยขั้นตอนที่เหมือนกันกล่าวคือ ทรีย่อยทางด้านซ้ายของ R จะถูกค้นหาก่อนทรีย่อยทางด้านขวามือ เพราะฉะนั้นความแตกต่างของแต่ละอัลกอริทึมก็คือ เวลาที่ทำการประมวลผลจากราก R สำหรับในอัลกอริทึมพรีออร์เดอร์ รากของ R จะถูกประมวลผลก่อนทรีย่อยในอัลกอริทึม สำหรับอินออร์เดอร์ รากของ R ถูกประมวลผลระหว่างการค้นหาในทรีย่อย และในอัลกอริทึมโพสออร์เดอร์ รากของ R จะถูกประมวลผลหลังจากการค้นหาทรีย่อย

อัลกอริทึมทั้ง 3 แบบบางครั้งจะเรียกว่า โหนด – ซ้าย – ขวา (Node – Left – Right NLK) และ ซ้าย – โหนด – ขวา (Left – Node – Right LNR) และ ซ้าย – ขวา – โหนด (Left – Right – Node LRN)

สังเกตว่าแต่ละอัลกอริทึมเป็นการกำหนดแบบเรียกตัวเอง เนื่องจากอัลกอริทึมมีการค้นหาทรีย่อยในลำดับที่กำหนด ซึ่งจะต้องใช้สแตกเมื่อมีการทำงานด้วยคอมพิวเตอร์

2.10 อัลกอริทึมในการเข้าถึงข้อมูลแบบใช้สแตก

สมมติไบนารีทรี T เก็บในหน่วยความจำที่ใช้โครงสร้างข้อมูลแบบลิงค์ TREE (Info, Left, Right, Root) สามารถแสดงได้ดังรูปที่ 2.10



รูปที่ 2.10 อัลกอริทึมในการเข้าถึงข้อมูลแบบใช้สแตก

ดังนั้นในหัวข้อนี้จะเป็นการพิจารณาการเข้าถึงข้อมูลของ T ทั้ง 3 แบบ ในลักษณะของการใช้สแตก ซึ่งเขียนเป็นโปรซีเยอร์แบบนอนรีเคอร์ซีฟได้ดังนี้

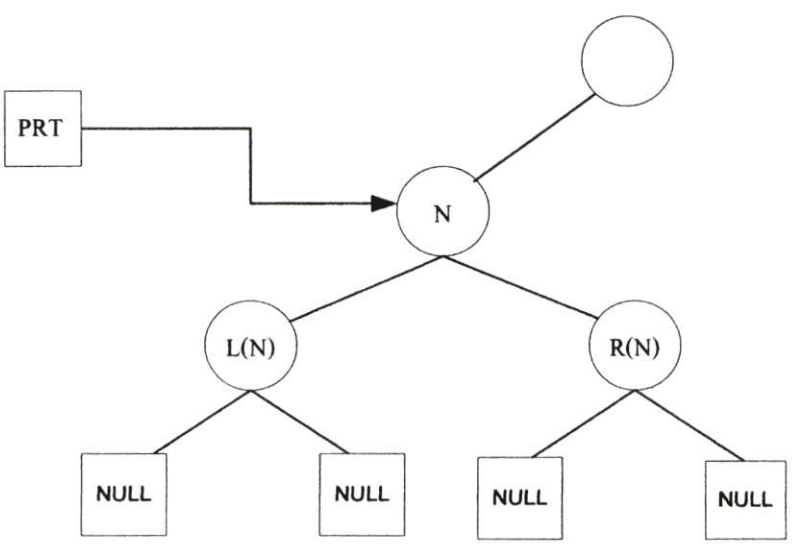
2.10.1 การเข้าถึงข้อมูลแบบพรีออร์เดอร์

อัลกอริทึมของการเข้าถึงข้อมูลแบบพรีออร์เดอร์นั้น จะใช้ตัวแปรพ้อยน์เตอร์ (Pointer) PTR สำหรับเก็บตำแหน่งของโหนด N ที่ติดต่ออยู่ ดังรูปที่ 2.11 เมื่อ L(N) เป็นโหนดลูกทางด้านซ้ายของโหนด N และ R(N) เป็นโหนดลูกทางด้านขวา อัลกอริทึมที่ใช้อาร์เรย์สแตก (Stack) ทำหน้าที่เก็บตำแหน่งของโหนด เพื่อนำไปใช้ต่อไป ซึ่งจากที่ได้กล่าวมาทั้งหมดนี้ สามารถนำมาเขียนเป็นลำดับขั้นตอนจะได้

Algorithm:เริ่มจากเก็บค่า NULL ลงใน STACK แล้วกำหนด PTR:= ROOT แล้วทำซ้ำขั้นตอนต่อไปนี้จนกระทั่ง PTR = NULL ซึ่งเท่ากับ while PTR $\neq$ NULL

(1) กระทำจาก PTR ลงไปทางซ้ายสุด โดยประมวลผลในแต่ละโหนดที่อยู่ในทางเดิน และเก็บโหนดลูกทางด้านขวา R(N) ลงในสแตกการเข้าถึงข้อมูลจะจบลง เมื่อมีการประมวลผลโหนด N ซึ่งไม่มีโหนดลูกทางด้านซ้ายค่า PTR จะเปลี่ยนไปจากการกำหนด PTR = LEFT(PTR) และการเข้าถึงข้อมูลสิ้นสุดลงเมื่อ LEFT [PTR] = NULL

(2) [ทำย้อนกลับ] โดยทำการดึงค่าจาก สแตก มาเป็นตัวกำหนดให้กับ PTR หากค่าของ PTR $\neq$ NULL แล้วกลับไปทำในขั้นตอน (1) หากเป็นอย่างอื่น ให้ออกจากโปรแกรม (Exit) โดยใช้การกำหนด NULL ลงในสแตก สำหรับเป็นเครื่องหมายจบ



รูปที่ 2.11 การเข้าถึงข้อมูลแบบพรีออร์เดอร์

2.10.2 การเข้าถึงข้อมูลแบบอินออร์เดอร์

อัลกอริทึมสำหรับการเข้าถึงข้อมูลแบบอินออร์เดอร์นี้ จะใช้ตัวแปรพ้อยน์เตอร์ PTR สำหรับเก็บตำแหน่งของโหนด N ที่ติดต่ออยู่ และใช้อาร์เรย์สแตก สำหรับเก็บตำแหน่งของโหนดที่จะประมวลผลต่อไป อัลกอริทึมนี้โหนดจะถูกประมวลผลหลังจากถูกดึงออกมาจาก สแตก ซึ่งสามารถนำมาเขียนเป็นลำดับขั้นตอนได้ดังนี้

Algorithm: เริ่มต้นจากการกำหนด NULL ลงใน สแตก (สำหรับแสดงการจบการทำงาน) และกำหนด PTR:= ROOT แล้วทำตามขั้นตอนต่อไปนี้ จนกว่า NULL จะถูกดึงออกมาจาก STACK

(1) กระทำลงไปทางด้านซ้ายจากราก โดย PTR จะทำหน้าที่เก็บแต่ละโหนด N ลงในสแตกและหยุดเมื่อโหนด N ไม่มีโหนดลูกทางด้านซ้าย

(2) [ทำย้อนกลับ] ดึงค่าจาก สแตก และประมวลผล ถ้าพบว่าเป็น NULL ก็จบการทำงาน ถ้าโหนด N มีโหนดลูกด้านขวาที่ถูกประมวลผล กำหนด PTR = R(N) (โดยใช้การกำหนด PTR:= RIGHT [PTR] และกลับไปทำในขั้นตอน (1)

2.10.3 การเข้าถึงข้อมูลแบบโพสท์ออร์เดอร์

อัลกอริทึมในการเข้าถึงข้อมูลแบบโพสท์ออร์เดอร์มีความยุ่งยากกว่าทั้ง 2 แบบที่ได้กล่าวมาเนื่องจากต้องมีการเก็บโหนด N ใน 2 ลักษณะที่แตกต่างกัน คือ จะเก็บ N หรือ – N ลงใน สแตก ซึ่งในทางปฏิบัติตำแหน่งของ N จะถูกเก็บลงใน สแตก และค่าของ –N คือ ค่าลบของ N ตัวแปร PTR จะถูกใช้สำหรับเก็บตำแหน่งของโหนด N ที่กำลังติดต่ออยู่ ดังในรูปที่ 2.11

Algorithm: เริ่มต้นเก็บ NULL ลงในสแตก (เป็นตัวบอกการจบการทำงาน) และทำการกำหนด PTR:= ROOT จากนั้นทำตามขั้นตอนต่อไปนี้ ซ้ำจนกระทั่งค่า NULL ถูกดึงออกมาจากสแตก

(1) ทำลงไปทางซ้ายจากราก PTR ที่แต่ละโหนด N ของทางผ่านเก็บ N ลงในสแตก และถ้า N มีโหนดลูกด้านขวา R (N) เก็บ – R (N) ลงในสแตก

(2) [ทำย้อนกลับ] โดยดึงค่าโหนดที่เป็นบวกออกจากสแตก ถ้าพบ Null ก็จะจบการทำงาน และถ้าค่าที่ดึงออกมาจากสแตกเป็นโหนดลบ นั่นคือ PTR:= - N กำหนด PTR:= N (โดยการกำหนด PTR:= PTR) แล้วกลับไปทำในขั้นตอนที่ (1) การประมวลผลที่ได้จะทำกับค่าโหนดที่ดึงออกมาจากสแตก และเป็นบวกเพียงเท่านั้น

2.11 ไบนารีเซิร์ททรี (Binary search trees)

สำหรับในส่วนนี้ จะเป็นการพิจารณาโครงสร้างข้อมูลที่สำคัญที่สุดในวิทยาการคอมพิวเตอร์ ซึ่งสามารถทำการค้นหาข้อมูลได้ โดยใช้เวลาในการค้นหา $f(n) = O(\log_2 n)$ และสามารถแทรกหรือลบอิลิเมนท์ได้ง่าย ๆ ซึ่งโครงสร้างจะแตกต่างกับโครงสร้างต่อไปนี้

(1) อาร์เรย์ที่มีการจัดเรียงเราสามารถค้นหา และ พบอิลิเมนท์ที่ต้องการได้โดยใช้เวลาประมาณ $f(n) = O(\log_2 n)$ แต่มีการแทรกและการลบอิลิเมนท์ที่ยุ่งยาก

(2) ลิงค์ลิสท์ สามารถลบและแทรกอิลิเมนท์ได้ง่าย แต่ก็มีความยุ่งยากในการค้นหาตำแหน่งของ อิลิเมนท์ที่ต้องการได้ เนื่องจากต้องใช้ลิเนียร์เซิรท์ที่มี $f(n) = O(n)$ นั่นเอง

ถึงแม้ว่าแต่ละโหนดในไบนารีเซิรท์ทรีนั้น จะสามารถเก็บเรคคอร์ดได้ก็ตาม แต่การจัดเรียงลำดับของโหนดในไบนารีทรี ก็ยังคงขึ้นอยู่กับฟิลด์ที่ต้องการ

สมมติให้ T เป็นไบนารีทรี จะเรียก T ว่าไบนารีเซิรท์ทรี (หรือ ไบนารีเซิรท์ทรี) ถ้าแต่ละโหนด N ของ T มีคุณสมบัติดังต่อไปนี้ ค่าใน N มีค่าสูงกว่าทุก ๆ ค่าที่อยู่ในทรีย่อยทางด้านซ้ายของ N และต่ำกว่า ทุก ๆ ค่าที่อยู่ในทรีย่อยทางด้านขวาของ N (ซึ่งจะมองได้ว่าคุณสมบัตินี้ ถ้ามีการเข้าถึงข้อมูลแบบอินออร์เดอร์ จะได้ข้อมูลที่มีการเรียงลำดับในแต่ละอิลิเมนท์ของ T)

2.11.1 การค้นหาและการแทรกในไบนารีเซิรท์ทรี

ในหัวข้อนี้จะกล่าวถึงการค้นหาและการแทรกโหนดลงในไบนารีทรี โดยจะแยกเป็นการค้นหาและการแทรก ตามลำดับ ส่วนการลบจะพิจารณาในหัวข้อต่อไป ซึ่งการเข้าถึงข้อมูลในทรี T จะใช้วิธีการดังที่ได้กล่าวมาแล้วในตอนต้น เมื่อ Item เป็นข้อมูลที่กำหนด อัลกอริทึมต่อไปนี้จะทำการค้นหาตำแหน่งของ Item ที่อยู่ในไบนารีเซิรท์ทรี หรือแทรก Item เป็นโหนดใหม่ในตำแหน่งที่เหมาะสมที่สุด

(1) เปรียบเทียบ Item กับโหนดราก N ของทรี

(1.1) ถ้า Item < N เปรียบเทียบไปทางโหนดลูกทางด้านซ้ายของ N

(1.2) ถ้า Item > N เปรียบเทียบไปทางโหนดลูกทางด้านขวาของ N

(2) ทำซ้ำขั้นตอนที่ (1) จนปรากฏเงื่อนไขดังนี้ เงื่อนไขหนึ่ง

(2.1) พบโหนด N ที่มี Item = N ในกรณีนี้การค้นหาถือว่าประสบผลสำเร็จ

(2.2) พบทรีย่อยว่าง หมายถึงการค้นหาไม่พบข้อมูล และทำการแทรก Item ลงในตำแหน่งของทรีย่อยที่ว่าง ลำดับการทำงานจะทำจากราก R ลงไปจนกระทั่งพบ Item ใน T หรือแทรก Item ลงไปเป็นโหนดสุดท้ายสุดของ T

2.11.2 ความซับซ้อนของอัลกอริทึมการค้นหา

เมื่อทำการค้นหา Item ในไบนารีเซิรท์ทรี T จำนวนการเปรียบเทียบจะจำกัดอยู่ด้วยค่าที่เป็นความลึกของทรี ดังนั้นเวลาที่ใช้จะเป็นอัตราส่วนของความลึกของทรี สมมติมีข้อมูล n รายการ A_1, A_2, ..., A_n ที่ต้องการแทรกลงในไบนารีเซิรท์ทรี T ตามลำดับ จากหัวข้อที่เคยกล่าวถึงมาแล้ว จะมีค่าเปอร์มิวเตชัน (Permutation) ของข้อมูล n รายการเป็น n! แสดงว่าความลึกโดยเฉลี่ยของทรีที่มี n! จะประมาณ c $\log_2$ n เมื่อ c = 1.4 ซึ่งจะได้เวลาในการค้นหา Item ในไบนารีทรี T ที่มีขนาด n อิลิเมนท์ คือ f (n) เป็นอัตราส่วนของ $\log_2$ n นั่นคือ O ($\log_2$ n)

2.11.3 การประยุกต์ใช้งานไบนารีเซิรท์ทรี

เมื่อมีข้อมูล n รายการ A_1, A_2, ..., A_n ในทรี ถ้าต้องการค้นหาและลบข้อมูลที่ซ้ำกันออกไป ดังนั้นสามารถทำได้ตามขั้นตอนดังนี้

อัลกอริทึม A: ตรวจสอบอิลิเมนท์จาก A_1 ถึง A_n (จากซ้ายไปขวา)

(1) แต่ละอิลิเมนท์ A_K เปรียบเทียบ A_K กับ A_1, A_2, ... , A_{K-1} ซึ่งการเปรียบเทียบ A_K กับอิลิเมนท์ที่อยู่ก่อนหน้า

(2) หากมีค่า A_K อยู่ระหว่าง A_1, A_2, ... , A_{K-1} ให้ลบ A_K ออกหลังจาก ตรวจสอบทุกอิลิเมนท์แล้ว จะได้ไม่มีข้อมูลที่ซ้ำกัน

2.11.4 การลบโหนดออกจากไบนารีเซิรท์ทรี

เมื่อ T เป็นไบนารีเซิรท์ทรี และต้องการลบ Item ออกจากทรี อัลกอริทึมของการลบจะใช้โปรซีเยอร์ทำการค้นหาตำแหน่งโหนด N ที่มีข้อความ Item ก่อน และตำแหน่งของโหนดแม่เป็น P(N) การลบ N จากทรี ขึ้นอยู่กับจำนวนโหนดลูกของ N ซึ่งมีอยู่ด้วยกัน 3 กรณี

กรณี 1 ถ้า N ไม่มีโหนดลูก ลบ N จาก T โดยการแทนตำแหน่งของ N ในโหนดแม่ P(N) ด้วยค่า NULL

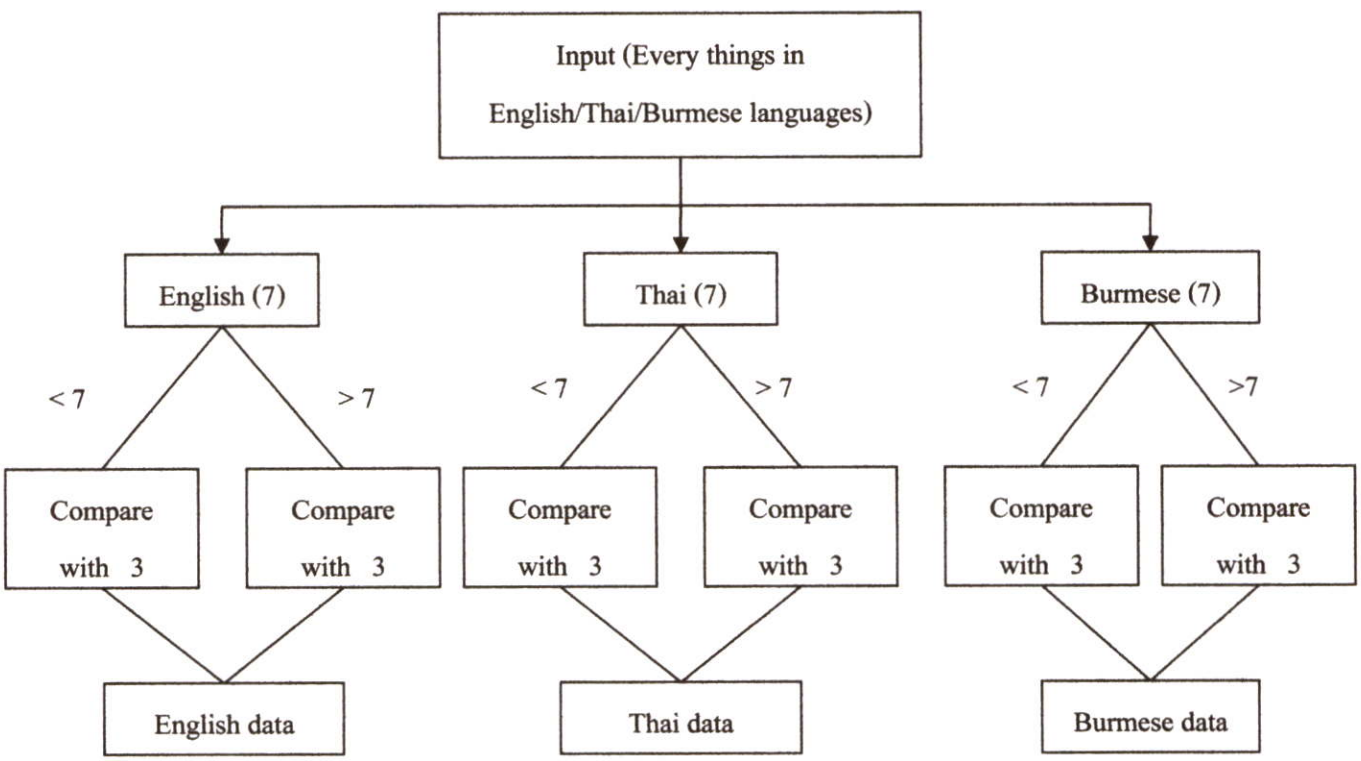
กรณี 2 ถ้า N มีโหนดลูกเพียงโหนดเดียว ลบ N โดยแทนตำแหน่งของ N ในโหนดแม่ P(N) ด้วยตำแหน่งของโหนดลูกของ N

กรณี 3 ถ้า N มีโหนดลูก 2 โหนด โดยกำหนด S(N) คือ อินออร์เดอร์ ซัคเซสเซอร์ของ N (ซึ่ง S(N) ไม่มีโหนดลูกด้านซ้าย) ทำการลบ N ออกจาก T โดยลบ S (N) จาก T ก่อน (โดยใช้กรณีที่ 1 หรือกรณีที่ 2) แล้วจึงแทนโหนด N ใน T ด้วยโหนด S(N)

สังเกตว่า ในกรณีที่ 3 จะยุ่งยากกว่า 2 กรณีแรก ทั้ง 3 กรณีหน่วยความจำที่ว่างของ N จะถูกส่งกลับไปยังลิสท์ AVAIL (ลิสท์ AVAIL คือ พอยเตอร์ของ ลิสท์)

บทที่ 3
การค้นหาข้อมูลไบนารีเซิร์ททรี

การทำนอร์มอลไลซ์ข้อมูล คือ การกระทำกับข้อมูลหรือการตระเตรียมข้อมูลก่อนที่จะนำเอาข้อมูลเหล่านั้น ผ่านเข้าสู่กระบวนการของโปรแกรม เพื่อให้ได้เอาต์พุตเป็นไปตามความต้องการ โดยการจัดเรียงข้อมูลมีอยู่ด้วยกัน 2 ขั้นตอนกล่าวคือในลำดับแรกจะทำการจองพื้นที่หรือจองหน่วยความจำเอาไว้ให้กับตัวแปรหรือพารามิเตอร์ต่าง ๆ ก่อนเพื่อเป็นการเพิ่มความเร็วในการประมวลผลให้เร็วขึ้นทั้งนี้การจองพื้นที่หรือหน่วยความจำ ให้กับพารามิเตอร์นั้นจะเป็นการลดกระบวนการในการรับส่งค่าพารามิเตอร์ลงจากสองขั้นตอนเหลือเพียง ขั้นตอนเดียวเท่านั้น เพราะเมื่อมีการรับค่ามาเก็บไว้ที่พารามิเตอร์แล้ว จะรู้ในทันทีเลยว่าค่าของพารามิเตอร์ตัวนี้จะต้องเอาไปเก็บไว้ที่ตำแหน่งใดของหน่วยความจำ โดยไม่จำเป็นต้องเสียเวลาไปหาตำแหน่งหรือหน่วยความจำที่ว่างอีกครั้งหนึ่งและนี่ก็เป็นเหตุผลว่าทำไมการจองพื้น

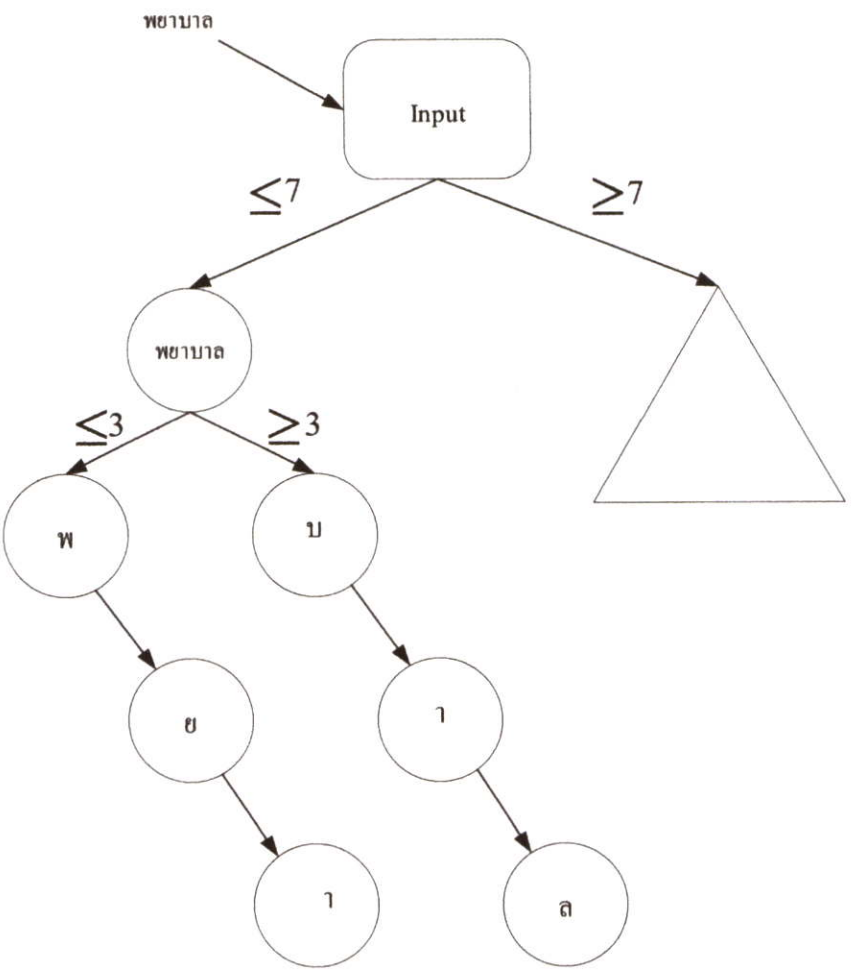


รูปที่ 3.1 ลำดับขั้นตอนในการทำงานของการนอร์มอลไลซ์ไบนารีเซิร์ททรี

ที่หรือหน่วยความจำจึงทำให้โปรแกรมสามารถประมวลผลได้เร็วขึ้น จากนั้นในขั้นตอนที่สองจะทำการจัดเรียงข้อมูลที่ถูกเก็บอยู่ในฐานข้อมูลเสียก่อน เพื่อให้ข้อมูลเหล่านี้ มีการเรียงเป็นลำดับขั้นตอนตามตัวอักษรเพราะฉะนั้นเมื่อถึงเวลาที่ต้องไปค้นหาข้อมูลเหล่านี้ในฐานข้อมูลก็จะ ทำให้การค้นหาเร็วขึ้นตาม โฟร์ชาร์ทที่ได้แสดงในรูปที่ 3.1 จากโฟร์ชาร์ท จะรับข้อมูลที่เป็น ภาษาไทย หรือภาษาอังกฤษ หรือ ภาษาพม่า เข้ามาตามที่ผู้ใช้ต้องการ จากนั้นจะนำข้อมูลที่รับเข้ามานี้ ซึ่งเป็นข้อมูลอินพุตมาทำการแบ่งแยกตามค่าเทรสโฮลที่ได้กำหนดเอาไว้ ซึ่ง ณ ขณะนี้ได้ทำการกำหนดค่าเทรสโฮลเอาไว้เท่ากับ 7 เพราะเป็นค่าทางสถิติที่ประมวลผลเร็วที่สุด จากนั้นเมื่อมีการแบ่งแยกข้อมูลอินพุตตามค่าเทรสโฮลแล้วนั้น ก็นำเอาข้อมูลนี้มาทำการเปรียบเทียบกับข้อมูลที่อยู่ในฐานข้อมูลทีละ 3 ตัวอักษร แล้วถึงจะมีการแสดงผลลัพธ์ออกมาในลำดับสุดท้าย ดั้งนั้นถ้าข้อมูลที่อยู่ในฐานข้อมูล ได้มีการจัดเรียงตามลำดับตัวอักษรเอาไว้แล้ว ก็จะทำให้การเปรียบเทียบตัวอักษรที่ได้กำหนดเอาไว้ในส่วนของการค้นหาคำศัพท์เป็นไปได้ง่ายและรวดเร็วกว่านั่นเอง

3.1 ตัวอย่างโครงสร้างต้นไม้ของคำต่าง ๆ

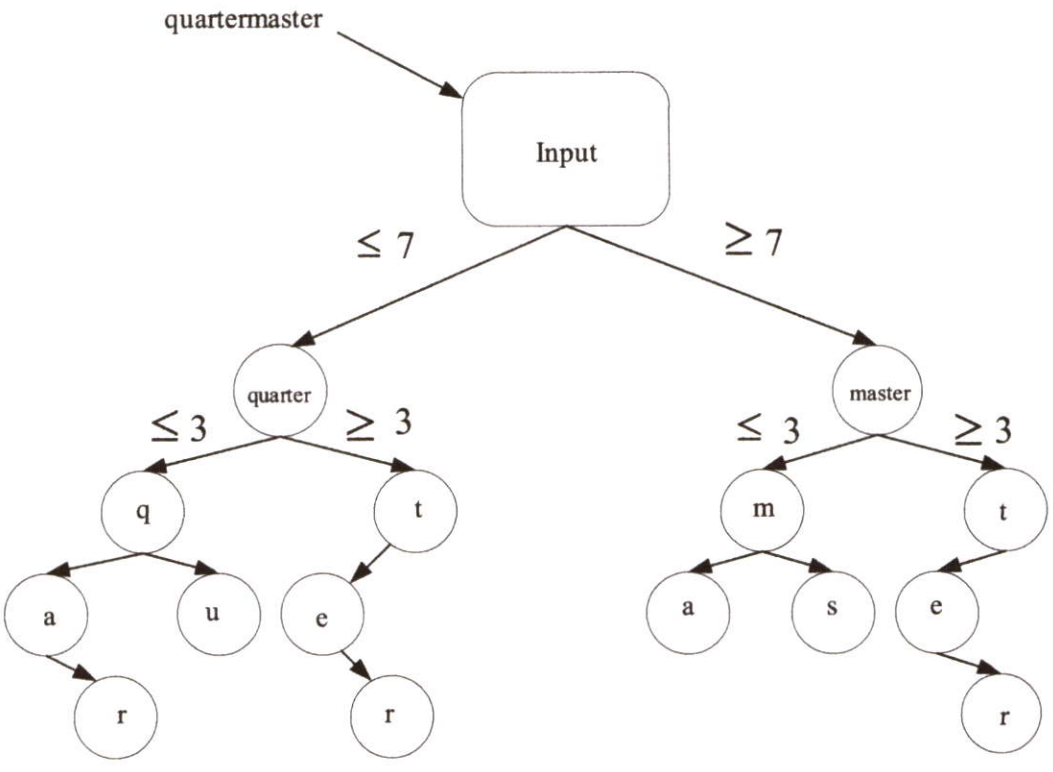
โครงสร้างต้นไม้ของคำว่า " พยาบาล "



รูปที่ 3.2 โครงสร้างต้นไม้ของคำว่า " พยาบาล "

เมื่อป้อนอินพุต คำว่า " พยาบาล " เข้ามาเพื่อต้องการค้นหา จะมีค่าน้อยกว่า 7 จะทำการค้นหาข้อมูลด้านซ้ายมือ และจะทำการแยกคำว่า " พยาบาล " ออกเป็นสองส่วน คือส่วนแรกถ้ามีตัวอักษรน้อยกว่าหรือเท่ากับ 3 ตัวคือ ตัวอักษร " พยา " จะนำไปค้นหาทางซ้าย และ ส่วนหลังตัวอักษร มากกว่าหรือเท่ากับ 3 ตัว คือคำว่า " บาล " จะนำไปค้นหาทางขวามือ

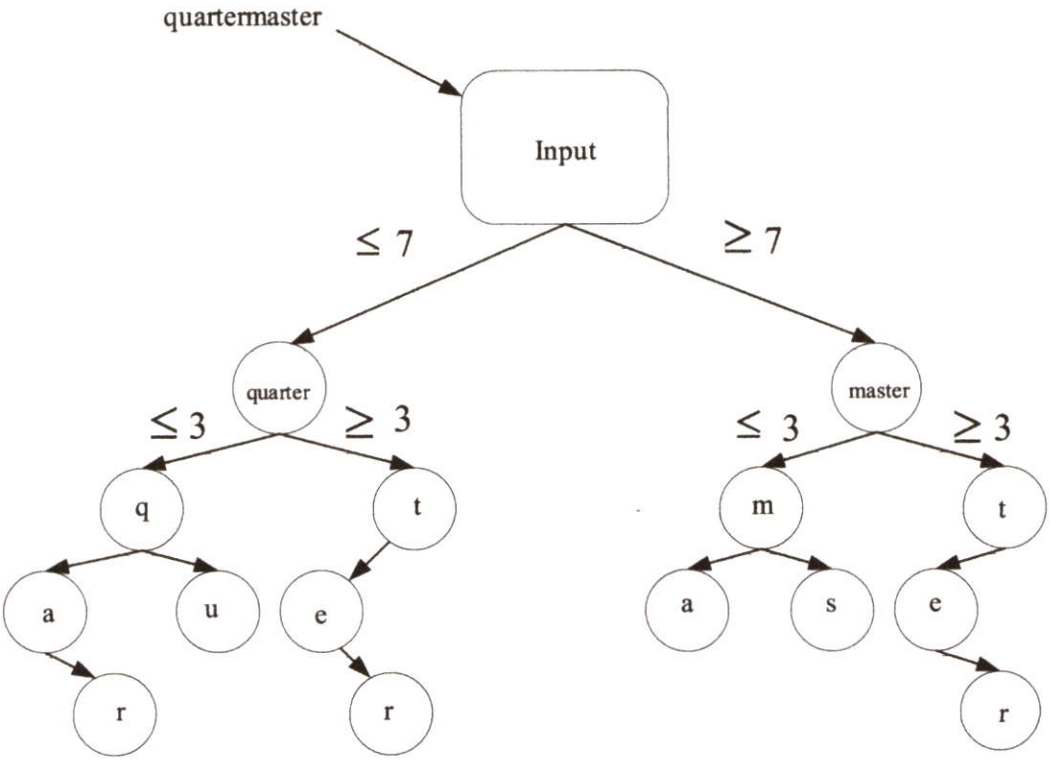
โครงสร้างต้นไม้ของคำว่า " quartermaster "



รูปที่ 3.4 โครงสร้างต้นไม้ของคำว่า "quartermaster "

เมื่อป้อนอินพุต คำว่า " quartermaster " เข้ามาเพื่อต้องการค้นหา จะมีตัวอักขระทั้งหมด 13 ตัว ค่าที่น้อยกว่า 7 คือ " quarter " จะทำการค้นหาข้อมูลด้านซ้ายมือ และ ค่าที่มากกว่า 7 คือ " master "จะทำการจะทำการค้นหาข้อมูลด้านขวามือ และทำการแบ่งตัวอักษรแยกคำว่า " quarter " ทางซ้ายออกเป็นสองส่วนคือ " quar " ไปค้นหาทางด้านซ้ายและ " ter " ไปค้นหาทางด้านขวา ส่วนอักขระ " master " ทางขวาจะแยกการค้นหาเป็นสองส่วนคือตัวอักษร " mas " จะแยกการค้นหาทางด้านซ้ายมือ และ ตัวอักษร " ter "จะแยกการค้นหาทางด้านขวามือ

โครงสร้างต้นไม้ของคำว่า " quartermaster "



รูปที่ 3.4 โครงสร้างต้นไม้ของคำว่า "quartermaster "

เมื่อป้อนอินพุต คำว่า " quartermaster " เข้ามาเพื่อต้องการค้นหา จะมีตัวอักขระทั้งหมด 13 ตัว ค่าที่น้อยกว่า 7 คือ " quarter " จะทำการค้นหาข้อมูลด้านซ้ายมือ และ ค่าที่มากกว่า 7 คือ " master "จะทำการจะทำการค้นหาข้อมูลด้านขวามือ และทำการแบ่งตัวอักษรแยกคำว่า " quarter " ทางซ้ายออกเป็นสองส่วนคือ " quar " ไปค้นหาทางด้านซ้ายและ " ter " ไปค้นหาทางด้านขวา ส่วนอักขระ " master " ทางขวาจะแยกการค้นหาเป็นสองส่วนคือตัวอักษร " mas " จะแยกการค้นหาทางด้านซ้ายมือ และ ตัวอักษร " ter "จะแยกการค้นหาทางด้านขวามือ

3.2 ความเร็วในการสืบค้น (BigOh)

3.2.1 ค่า **BigOh** ในการสืบค้นข้อมูล

- มีการใช้ดัชนีอักษรภาษาพม่า 34 อักษร
- มีการจำแนกแฟ้มคำตามรหัสอักษร(ภาษาพม่า) สำหรับแต่ละรายอักษรแตกต่างกันไปตามรายละเอียดจำนวนคำ
- แบ่งไฟล์โดยพิจารณาจากลำดับคำตำแหน่งที่สองเทียบสัดส่วน จำนวนคำให้อยู่ในอัตรา 1:1 (Optimized)ในกรณีที่ไม่เกิน 2000 คำ และ1:10 (Worst Case) ในกรณีที่มีจำนวนคำระหว่าง 2000 - 20000 คำ (สูงสุด2000 คำ ต่อแฟ้ม) ตัวอย่างเช่น กรณีที่คำซึ่งนำหน้าด้วยอักษรa ในภาษาพม่ามีอักษรตามในลำดับที่สองเป็นจำนวนใน ฐานข้อมูลทั้งสิ้น345 คำ (อักษรตามลำดับที่สองแตกต่างกันออกไป) สามารถจัดแฟ้มเป็นแฟ้มเดียว ซึ่งจะทำให้ได้สัดส่วนคำในแฟ้มจำนวนคำทั้งหมดเป็น1:1 ตัวอย่างในกรณี ซึ่งคำนำหน้าด้วยอักษรหนึ่งมีจำนวน ทั้งหมด 4,821 คำ แบ่งออกเป็น 3 แฟ้ม แยกลำดับด้วยอักษรตำแหน่งที่ 1 และ 2 ซึ่งจะทำให้ได้ จำนวนคำในแฟ้ม: จำนวนคำทั้งหมด เป็น 1:3 เป็นต้น
- เก็บเลขแฟ้มที่จำแนกคำเหล่านั้นสำหรับการสืบค้น แบบเร็วเมื่อสั่งสืบ ค้น ระบบจะค้นหาโดยใช้เพียงอักษรสองตัวแรก และ มุ่งไปที่ เลขแฟ้มและเปิดแฟ้มนั้น ๆ ขึ้นมาแสดงผล
- กรณีที่ไม่พบคำที่ตรงทุกอักษร จะแสดงผลตามคำที่ให้ผลใกล้เคียงที่สุด โดย เป็นคำแรกที่พบซึ่งมีจำนวนอักษรที่ หาไม่พบต่ำที่สุด เช่นหากเปรียบเทียบเป็น ภาษาอังกฤษคำที่ต้องการสืบค้นเป็น ABACKA แต่มีคำใกล้เคียงสองคำ คือABAC และ ABACKจะให้แสดง ABACK ออกมา

3.2.2 **BigOh** สำหรับการสืบค้นแบบนี้ สรุปได้เป็น

N = จำนวนอักษรที่แบ่งกลุ่มในลำดับแรก

K = จำนวนแฟ้มที่ถูกแบ่งสำหรับแต่ละกลุ่ม

W = จำนวนคำทั้งหมด

คำนวณ BigOh ของการสืบค้นได้เป็น (W/NK)

บทที่ 4
สถาปัตยกรรมของระบบ

4.1 ข้อมูลในระบบ

ฐานข้อมูลเป็นการจัดเก็บข้อมูลอย่างเป็นระบบ ทำให้ผู้ใช้สามารถใช้ข้อมูลที่เกี่ยวข้องในระบบงานต่าง ๆ ร่วมกันได้ โดยที่จะไม่เกิดความซ้ำซ้อนของข้อมูล และยังสามารถหลีกเลี่ยงความขัดแย้งของข้อมูลด้วย อีกทั้งข้อมูลในระบบจะมีความถูกต้องเชื่อถือได้ และ เป็นมาตรฐานเดียวกัน โดยจะมีการกำหนดระบบความปลอดภัยของ ข้อมูลขึ้น

นับได้ว่าปัจจุบันนี้เป็นยุคของสารสนเทศและเป็นที่ยอมรับกันว่า ระบบสารสนเทศเป็นข้อมูลที่ผ่านการกลั่นกรองอย่างเหมาะสม สามารถนำมาใช้ประโยชน์อย่างมากมาย ไม่ว่าจะเป็นการนำมาใช้งานในด้านธุรกิจ การบริหาร และกิจการอื่น ๆ สำหรับองค์กรที่มีข้อมูลปริมาณมาก ๆ จะพบความยุ่งยากลำบากในการจัดเก็บข้อมูล ตลอดจนการนำข้อมูลที่ต้องการออกมาใช้ให้ทันต่อเหตุการณ์ ดังนั้นคอมพิวเตอร์จึงถูกนำมาใช้เป็นเครื่องมือช่วยในการจัดเก็บข้อมูล การประมวลผลข้อมูล ซึ่งทำให้ระบบการจัดเก็บข้อมูลเป็นไปได้อย่างสะดวก ทั้งนี้โปรแกรมแต่ละโปรแกรมจะต้องสร้างวิธีควบคุมและจัดการกับข้อมูลขึ้นเอง ดังนั้นฐานข้อมูลจึงเข้ามามีบทบาทสำคัญเป็นอย่างมาก โดยเฉพาะระบบงานต่าง ๆ ที่ใช้คอมพิวเตอร์ การออกแบบและพัฒนาระบบฐานข้อมูลจึงต้องคำนึงถึงการควบคุมและการจัดการความถูกต้อง ตลอดจนประสิทธิภาพในการเรียก ใช้ข้อมูลด้วย

4.2 ข้อมูลที่ใช้ในสถาปัตยกรรมของระบบ

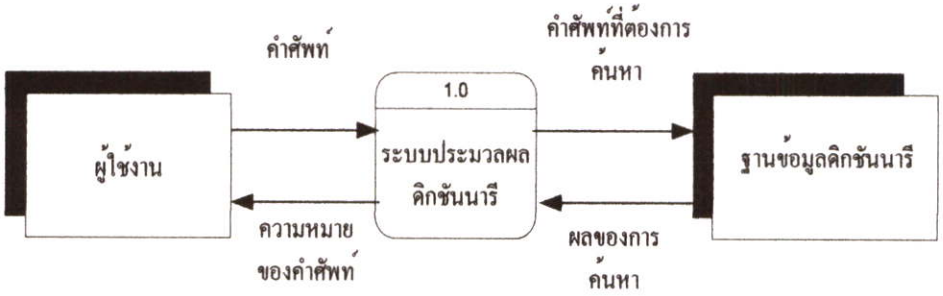
ข้อมูลดิบที่นำมาจัดรวบรวมเพื่อจัดทำระบบพจนานุกรม 3 ภาษานี้ นำมาจาก ภาษาไทย ภาษาพม่า และ ภาษาอังกฤษ ในแต่ละภาษาจะมีคำที่มีความหมายเฉพาะตามภาษานั้น ๆ แต่อาจมีคำหลายคำในแต่ละภาษาที่ถูกใช้แทนความหมายของสิ่งเดียว หรือ กล่าวได้ว่า สิ่งของ หรือ อาการนามเพียงหนึ่งคำอาจมีคำที่ถูกใช้อ้างอิงถึงได้เกินกว่า หนึ่งคำ ดังนั้นการจัดกลุ่มข้อมูล เพื่อนำมาเรียบเรียงเป็น พจนานุกรมสามภาษาดังกล่าวจึงต้องจัดกลุ่มข้อมูลเพื่อการเปรียบเทียบผ่าน พร้อมกับจัดโครงสร้างข้อมูล ให้เกิดประสิทธิภาพต่อการสืบค้นคำผ่านภาษาทั้งสามส่วน โครงสร้างของข้อมูลดังกล่าว ประกอบด้วย

- ภาษาไทย
- ภาษาอังกฤษ
- ภาษา พม่า
- เสียง
- รูปภาพ

จากนั้นจะนำส่วนประกอบเหล่านี้มาจัดกลุ่มข้อมูลเป็นตารางและจัดเก็บเป็นระบบฐานข้อมูล เพื่อทำการสืบค้นคำศัพท์ ต่อไป

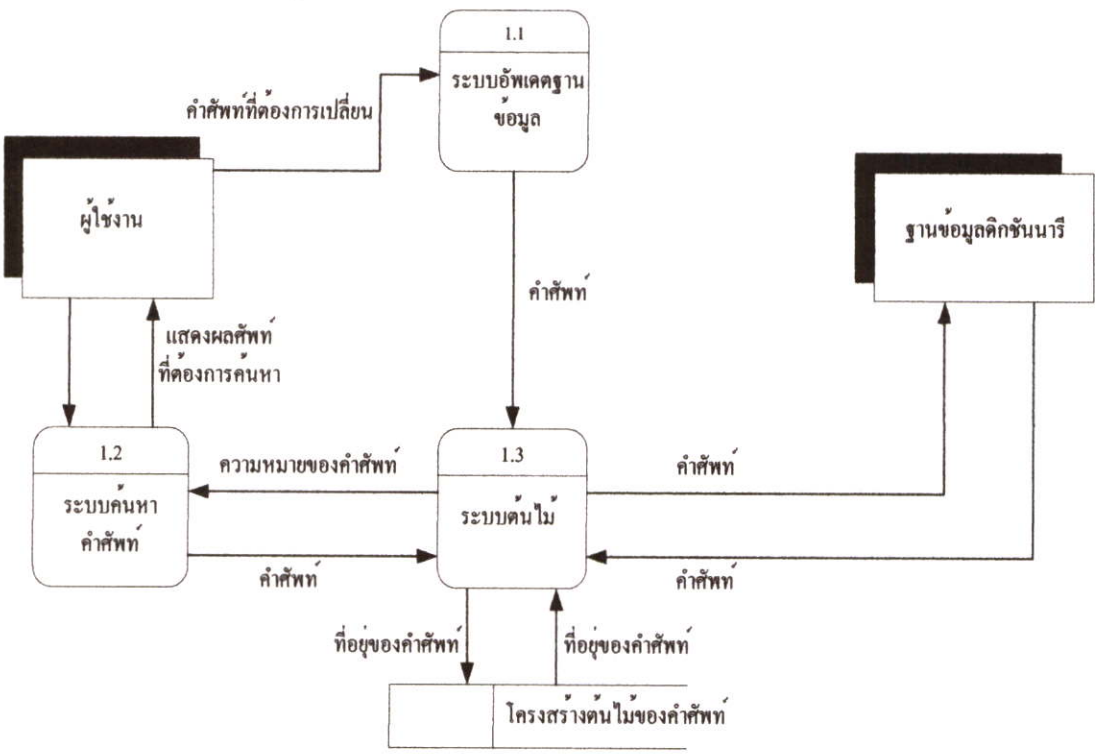
4.3 ระบบดิกชันนารี

4.3.1 Context diagram



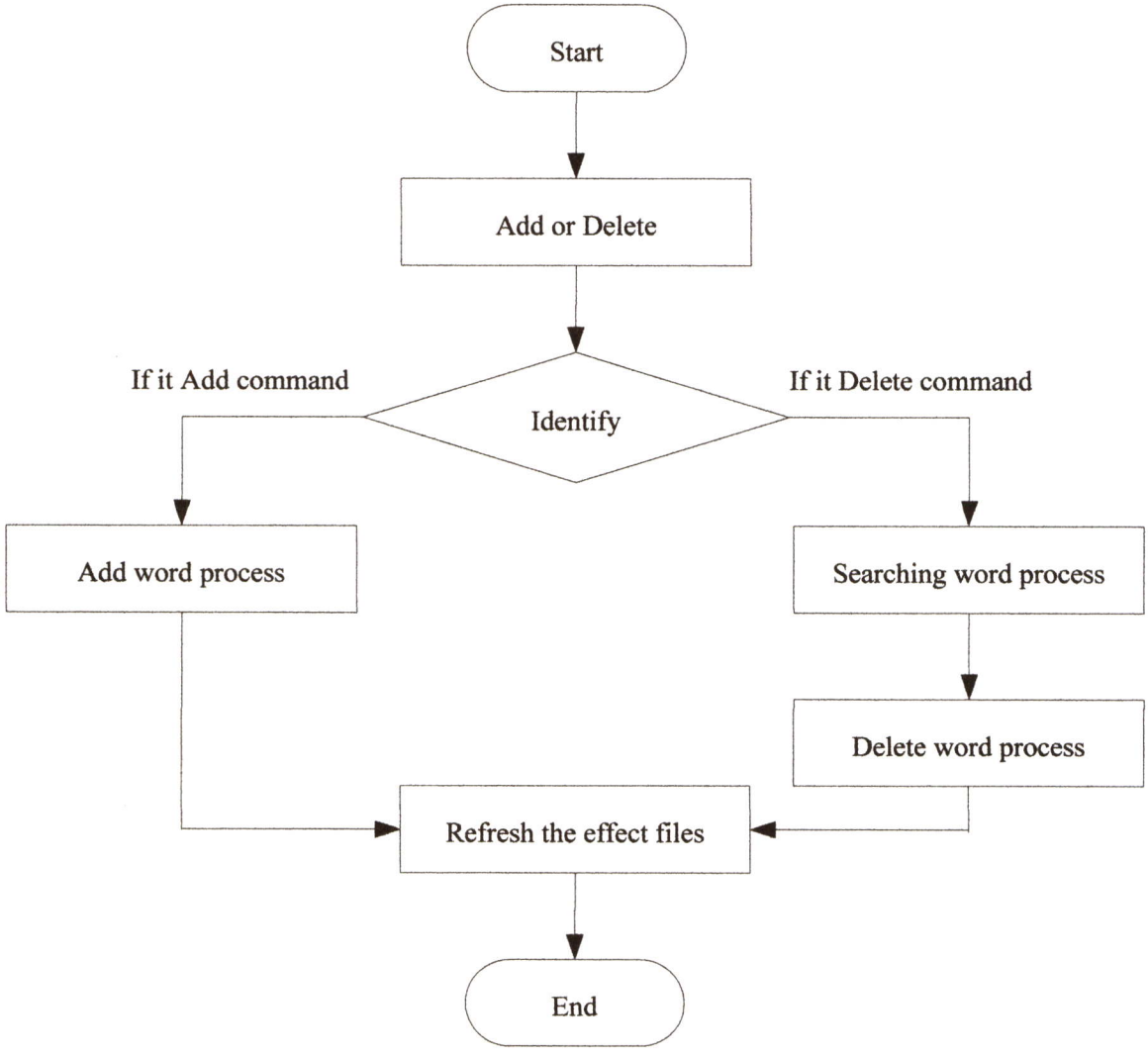
รูปที่ 4.1 Context diagram ของระบบ

4.3.2 Data flow diagram level 1



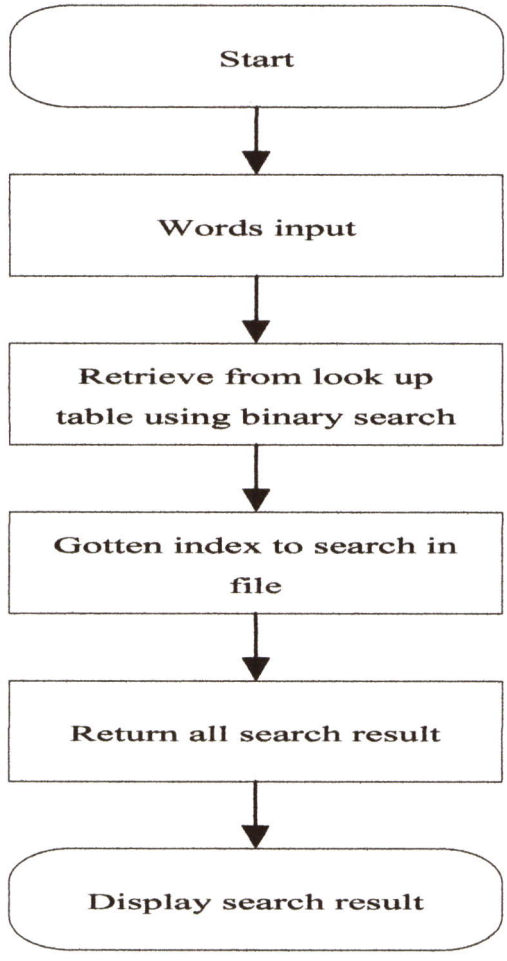
รูปที่ 4.2 Data flow diagram level 1 ของระบบ

ระบบอัพเดตฐานข้อมูล



รูปที่ 4.3 การเพิ่มและลดคำศัพท์

ระบบค้นหาคำศัพท์



รูปที่ 4.4 การค้นหาคำศัพท์

4.4 ตารางฐานข้อมูล

ชื่อเขตข้อมูล	ประเภท	ความยาว
คีย์หลัก (PK)	ตัวเลข	10
รหัสประเภทภาษา	ตัวเลข	10
ภาษาไทย	ข้อความ	256
ภาษาอังกฤษ	ข้อความ	256
ภาษาพม่า	ข้อความ	256
รูปภาพ	ไฟล์ (.jpeg)	
เสียงภาษาพม่า	ไฟล์ (.wav)	

บทที่ 5

ผลการทดลอง

5.1 วิธีดำเนินการทดลอง

สำหรับขั้นตอนในการดำเนินการทดลองนั้น จะเริ่มต้นจากการป้อนข้อมูลที่เป็นอินพุต หรือ คำศัพท์ที่ต้องการอยากรู้ความหมายเข้าไป ซึ่งคำศัพท์ที่ว่านี้อาจจะเป็นภาษาไทย ภาษาอังกฤษ หรือ ภาษาพม่าก็ได้ จากนั้นในลำดับต่อไปตัวโปรแกรมจะทำหน้าที่ในการเข้าไปค้นหาข้อมูลในฐานข้อมูลที่เก็บไว้ โดยวิธีที่ใช้ในการเข้าไปค้นหาข้อมูลนั้นก็คือไบนารีเซิร์ททรี (Binary Search Tree) นั่นเอง สำหรับกระบวนการทำงานต่าง ๆ ของอัลกอริทึมไบนารีเซิร์ททรี ได้ทำการอธิบายเอาไว้ในบทที่ผ่านมาเป็นที่เรียบร้อยแล้วสำหรับรายละเอียดต่าง ๆ กัน และในลำดับสุดท้ายเอาต์พุตที่จะแสดงผลออกมาจะเป็นความหมายของคำศัพท์นั้น ไม่ว่าจะเป็นภาษาไทย ภาษาอังกฤษ หรือภาษาพม่า รวมทั้งแสดงภาพและเสียงประกอบด้วย นอกเหนือจากนี้ในวิทยานิพนธ์นี้ได้ทดสอบความเร็วในการค้นหาคำศัพท์ ด้วยการจับเวลาที่ใช้ในการประมวลผลของการหาความหมายของคำศัพท์โดยการสุ่มจำนวน 500 คำ จากคำศัพท์ทั้งหมด 10,000 15,000 และ 20,000 คำ ตามลำดับ แล้วนำเวลาที่ใช้ในการค้นหาคำศัพท์ 500 คำ มา พล็อตกราฟ เพื่อพิจารณาผล และได้เปรียบเทียบความเร็วที่ใช้ในการหาคำศัพท์กับวิธีอื่น ๆ เช่น N-best [8] และ Lattice [12]

5.2 ผลการทดลอง

ผลการทดลองแสดงอัตราความเร็วในการค้นหาคำศัพท์ โดยการเปรียบเทียบ ของอัลกอริทึม แบบไบนารีเซิร์ททรี, N-best และ Lattice จากคำศัพท์ 10,000 15,000 และ 20,000 คำ ตามลำดับ โดยแสดงใน ตารางที่ 5.1, 5.2 และ 5.3 ตามลำดับ และนำอัตราความเร็วทั้ง 3 อัลกอริทึม มา พล็อตกราฟได้ดังแสดงในรูปที่ 5.1, 5.2 และ 5.3 ตามลำดับ

ตารางที่ 5.1 เป็นตารางเปรียบเทียบอัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์เป็นภาษาไทย ด้วยวิธีNormalize binary search tree, N-best และLatticeจากคำศัพท์ในฐานข้อมูลทั้งหมด10,000 คำ ทำการสุ่มคำศัพท์มา 500 คำ แล้วนำตัวอย่างคำศัพท์เพียง 25 คำมาเปรียบเทียบกันทั้ง 3 อัลกอรึทึม จะเห็นได้ว่าวิธี ของ Normalize binary search tree ค้นหาคำศัพท์ เร็วกว่า วิธีการของ N-best และ Lattice

ตารางที่ 5.1 อัตราเร็ว ที่ใช้ในการค้นหาคำศัพท์ทั้งหมด 10,000 คำ

Words	Algorithm types / Searching speed(sec) for 10000 words		
	Binary search tree	N-best[8]	Lattice[12]
ฝนตก	0.610461	2.954109	4.7094896
ห้วย	0.54256	2.7198567	5.4915571
น้ำพุ	0.559521	2.5673203	4.8844230
ถ่านหิน	0.583145	2.9418508	5.2666781
เหล็กกล้า	0.536886	2.8319651	4.6413967
ช้าง	0.58316	3.0922313	5.5166375
เลือด	0.513327	3.3352651	5.0053968
งูเหลือม	0.712808	2.6838855	5.7760185
ลิ้นจี่	0.508372	3.2202133	5.7658482
มะไฟ	0.621049	2.1594342	5.3497006
สีดำ	0.405535	3.3696203	5.1944839
เที่ยงคืน	0.574122	2.3071708	5.8396306
แรม ๑๕ ค่ำ	0.556344	2.2784940	5.7146066
มิถุนายน	0.671944	3.2136128	5.4318566
ฝ่า	0.738968	3.0059589	5.7087589
รวงข้าว	0.73696	2.1047781	5.4563640
น้า(ชาย,หญิง)	0.669181	1.9176350	4.9471529
การแต่งงาน	0.518202	3.2408469	4.8635614
แก้ม	0.510817	2.1987071	4.9459097
ลิ้น	0.902436	2.3480845	5.2545264
ปัสสาวะ	0.735026	2.8921638	5.5633811
ความนุ่มนวล	0.684984	2.3266128	4.8948642
การได้กลิ่น	0.512808	2.6038364	5.7415936
รถพยาบาล	0.680319	1.9971716	5.3089159
ยา	0.511544	3.3825024	4.9926616
เฉลี่ย	0.607	2.908	5.290

ตารางที่ 5.2 เป็นตารางเปรียบเทียบอัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์เป็นภาษาไทย ด้วยวิธี Normalize binary search tree, N-best และ Lattice จากคำศัพท์ในฐานข้อมูลทั้งหมด 15,000 คำ ทำการสุ่มคำศัพท์ มา 500 คำ แล้วนำตัวอย่างคำศัพท์เพียง 25 คำ มาเปรียบเทียบกันทั้ง 3 อัลกอรึทึม จะเห็นได้ว่าวิธี ของ Normalize binary search tree ค้นหาคำศัพท์ เร็วกว่า วิธีการของ N-best และ Lattice

ตารางที่ 5.2 อัตราเร็ว ที่ใช้ในการค้นหาคำศัพท์ทั้งหมด 15,000 คำ

Words	Algorithm types / Searching speed(sec) for 15000 words		
	Binary search tree	N-best [8]	Lattice [12]
atmosphere	0.994418	4.344605	6.582141
mound	0.830197	4.381156	6.231691
eddy	0.827575	3.998837	6.434125
limestone	0.869863	3.333451	6.034693
bird	0.75685	3.446791	6.252394
mongoose	0.862679	2.906534	5.797089
snipe	0.762696	3.722635	6.575575
fox	0.826492	3.108067	6.972803
durian	0.761252	4.324187	6.465616
lemon	0.760667	3.880634	6.709154
seed	0.611258	3.282711	5.759952
daytime	0.868065	3.914674	6.135604
waning moon	0.852976	3.009099	7.023323
October	0.776035	3.335856	6.797866
magnifying glass	0.836791	4.273422	6.314409
dynamite	0.861667	3.798331	6.549223
kindling	0.77169	2.903863	7.047263
quicklime	0.770814	4.094552	5.430201
counterfeit	0.765866	3.694694	6.626311
strength	1.400378	2.754168	5.410591
knock the door	0.863513	2.893552	6.213824
university	0.776925	3.653677	5.970997
restaurant	0.771251	3.696786	7.223607
stomach	0.954249	3.203905	5.333028
nationality	0.76439	3.592785	6.989498
เฉลี่ย	0.835	3.581	6.355

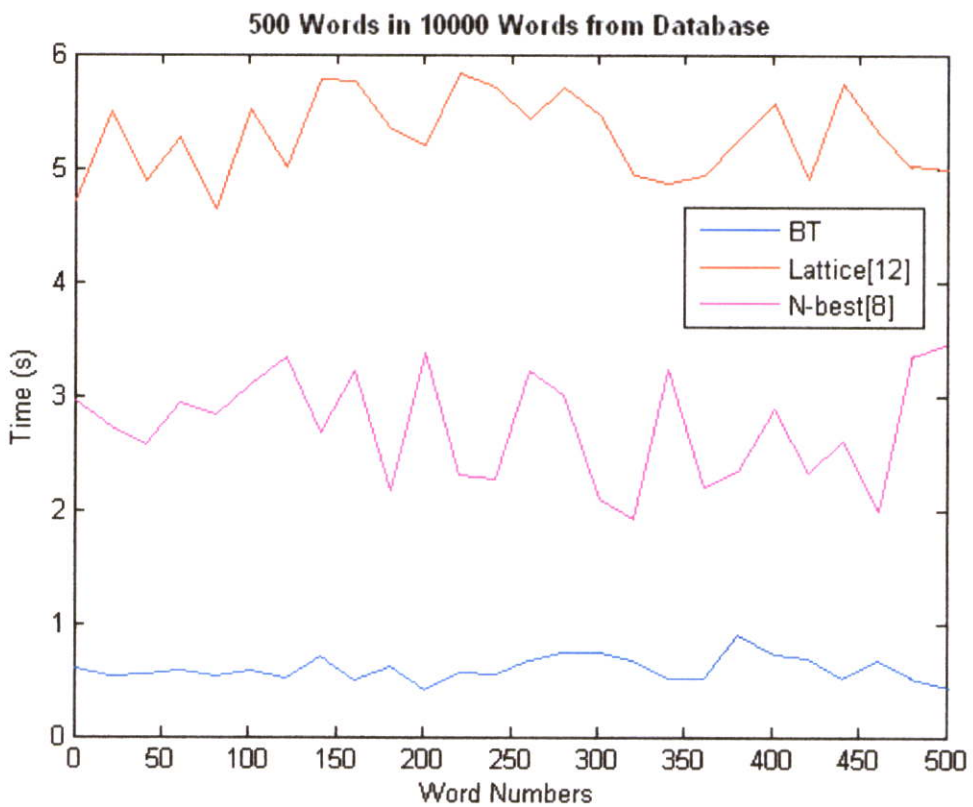
ตารางที่ 5.3 เป็นตารางเปรียบเทียบอัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์เป็นภาษาอังกฤษ ด้วยวิธี Normalize binary search tree, N-best และ Lattice จากคำศัพท์ในฐานข้อมูลทั้งหมด 20000 คำ ทำการสุ่มคำศัพท์ มา 500 คำ แล้วนำมาตัวอย่างคำศัพท์เพียง 25 คำเปรียบเทียบกันทั้ง 3 อัลกอรึทึมจะเห็นได้ว่าวิธี ของ Normalize binary search tree ค้นหาคำศัพท์ เร็วกว่าวิธีการของ N-best และLattice

ตารางที่ 5.3 อัตราเร็ว ที่ใช้ในการค้นหาคำศัพท์ทั้งหมด 20,000 คำ

Words	Algorithm types / Searching speed(sec) for 20000 words		
	Binary search tree	N-best[8]	Lattice[12]
universe	1.124473	4.506913	8.010232
crayon	1.622875	4.496804	6.716049
seashore	1.254793	5.509481	7.392316
pigeon	1.268568	4.680469	7.174768
butterfly	1.477058	4.497035	7.904338
become dark	1.603254	4.038184	7.671774
frequently	1.450561	5.866964	7.121641
go together	1.585331	5.112741	6.333306
proclaim	1.473398	6.329536	7.778532
ambassador	1.476187	5.164985	7.100466
husband	1.234735	5.046623	7.407778
previous	1.605868	6.115553	7.725486
muscle	1.435976	5.312881	7.959263
umbrella	1.462829	4.506618	7.628773
schedule	1.576998	5.680343	6.617279
rest room	1.618354	6.095296	7.030271
light bulb	1.483568	4.049098	7.983845
Quartermaster general department	1.153032	5.703192	7.950427
Director of operations	1.473669	4.948702	7.038486
faculty of education	2.286492	6.079497	7.908569
Chief of staff	1.274965	5.257032	6.404204
marines	1.127026	5.773678	6.935162
independent country	1.497168	5.072239	7.763699
watermelon	1.499059	4.761543	6.317753
gentleness	1.129826	4.474134	6.550003
เฉลี่ย	1.477	5.163	7.296

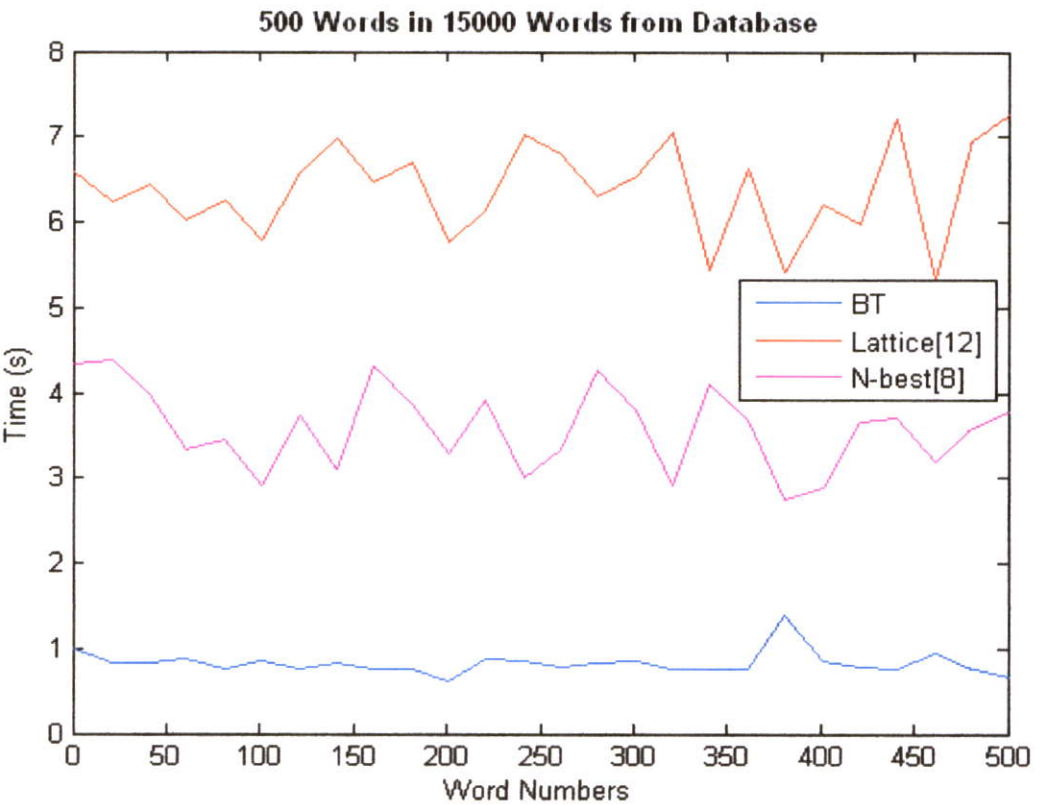
จากผลการทดลองในตาราง 5.1, 5.2 และ 5.3 จะเห็นได้ว่าอัตราเร็วที่ใช้ในการค้นหาคำศัพท์ด้วยวิธี นอร์มอลไลซ์ไบนารีเซริชทรี จากการสุ่มคำศัพท์มา 500 คำเลือกเพียง 25 คำ จากฐานข้อมูลคำศัพท์ทั้งหมด 10,000, 15,000 และ 20,000 คำ จะใช้เวลาค้นหาเฉลี่ย 0.607, 0.835 และ 1.477 วินาทีตามลำดับ

รูปที่ 5.1 เป็นรูปกราฟเปรียบเทียบอัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์ด้วยวิธี Normalize binary search tree, N-best และ Lattice จากคำศัพท์ในฐานข้อมูลทั้งหมด 10,000 คำ ทำการสุ่มคำศัพท์ มา 500 คำ แล้วนำมาเปรียบเทียบกันทั้ง 3 อัลกอริทึมจะเห็นได้ว่าวิธี ของ Normalize binary search tree ค้นหาคำศัพท์ได้ เร็วกว่า วิธีการของ N-best และ Lattice



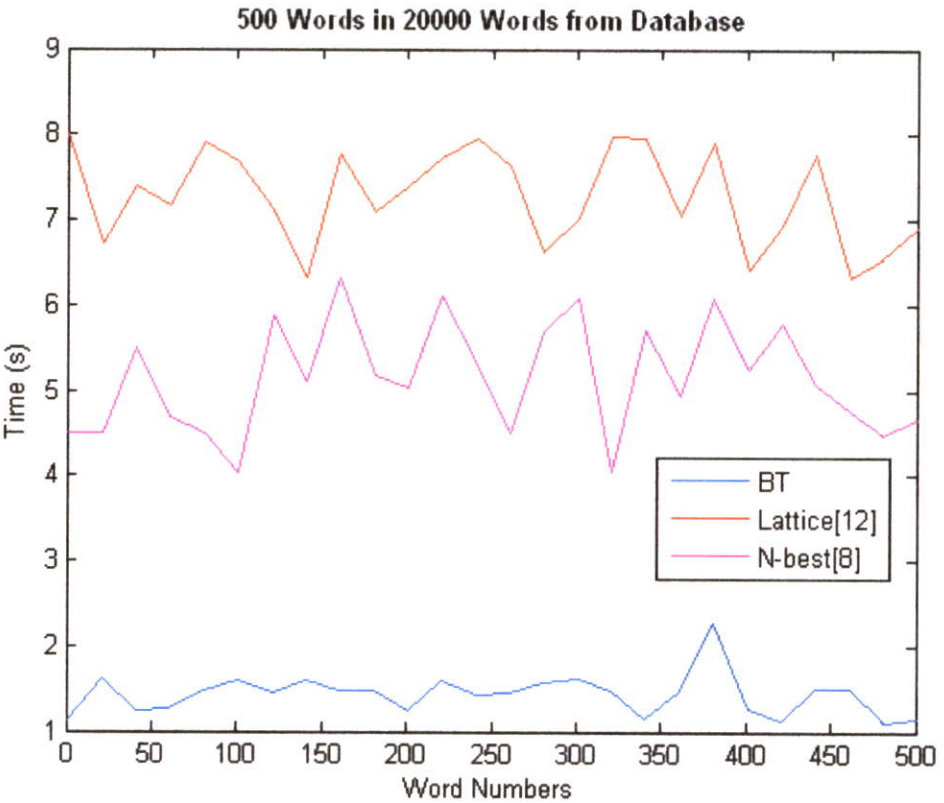
รูปที่ 5.1 อัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์ทั้งหมด 10,000 คำ

รูปที่ 5.2 เป็นรูปกราฟเปรียบเทียบอัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์ด้วยวิธี Normalize binary search tree, N-best และ Lattice จากคำศัพท์ในฐานข้อมูลทั้งหมด 15,000 คำ ทำการสุ่มคำศัพท์ มา 500 คำ แล้วนำมาเปรียบเทียบกันทั้ง 3 อัลกอริทึมจะเห็นได้ว่าวิธี ของ Normalize binary search tree ค้นหาคำศัพท์ ได้เร็วกว่า วิธีการของ N-best และ Lattice



รูปที่ 5.2 อัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์ทั้งหมด 15,000 คำ

รูปที่ 5.3 เป็นรูปกราฟเปรียบเทียบอัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์ด้วยวิธี Normalize binary search tree, N-best และ Lattice จากคำศัพท์ในฐานข้อมูลทั้งหมด 20,000 คำ ทำการสุ่มคำศัพท์ มา 500 คำ แล้วนำ มาเปรียบเทียบกันทั้ง 3 อัลกอรึทึม จะเห็นได้ว่าวิธี ของ Normalize binary search tree ค้นหาคำศัพท์ ได้เร็วกว่า วิธีการของ N-best และ Lattice



รูปที่ 5.3 อัตราความเร็ว ที่ใช้ในการค้นหาคำศัพท์ศัพท์ทั้งหมด 20,000 คำ

จากผลการทดลองจะเห็นว่าในรูปที่ 5.1, 5.2 และ 5.3 แสดงอัตราความเร็วที่ใช้ในการค้นหาคำศัพท์จำนวน 500 คำ จากคำศัพท์ทั้งหมด 10,000, 15,000 และ 20,000 คำ ตามลำดับ วิธีนอร์มอลไลซ์ไบนารีเซิร์ททรี (Normalize Binary Search Tree) ใช้เวลาในการค้นหาเฉลี่ย 0.725, 0.963 และ 1.586 วินาที จากคำศัพท์ทั้งหมด 10,000, 15,000 และ 20,000 คำ ตามลำดับ แต่ถ้าหากใช้วิธีการค้นหาคำศัพท์ด้วยวิธี N-best[8] จะต้องใช้เวลาในการค้นหาเฉลี่ย 3.168, 3.725 และ 5.328 วินาที ตามลำดับ และวิธี Lattice[12] จะต้องใช้ เวลาในการค้นหาเฉลี่ย 5.412, 6.546 และ 7.436 วินาที ตามลำดับ จะเห็นได้ว่าวิธีนอร์มอลไลซ์ไบนารีเซิร์ททรี สามารถค้นหาคำศัพท์ได้เร็วกว่าการค้นหาคำศัพท์ด้วยวิธีของ N-best[8] และ Lattice[12]

5.3 การแสดงผลเมื่อคำศัพท์นั้นมีภาพ

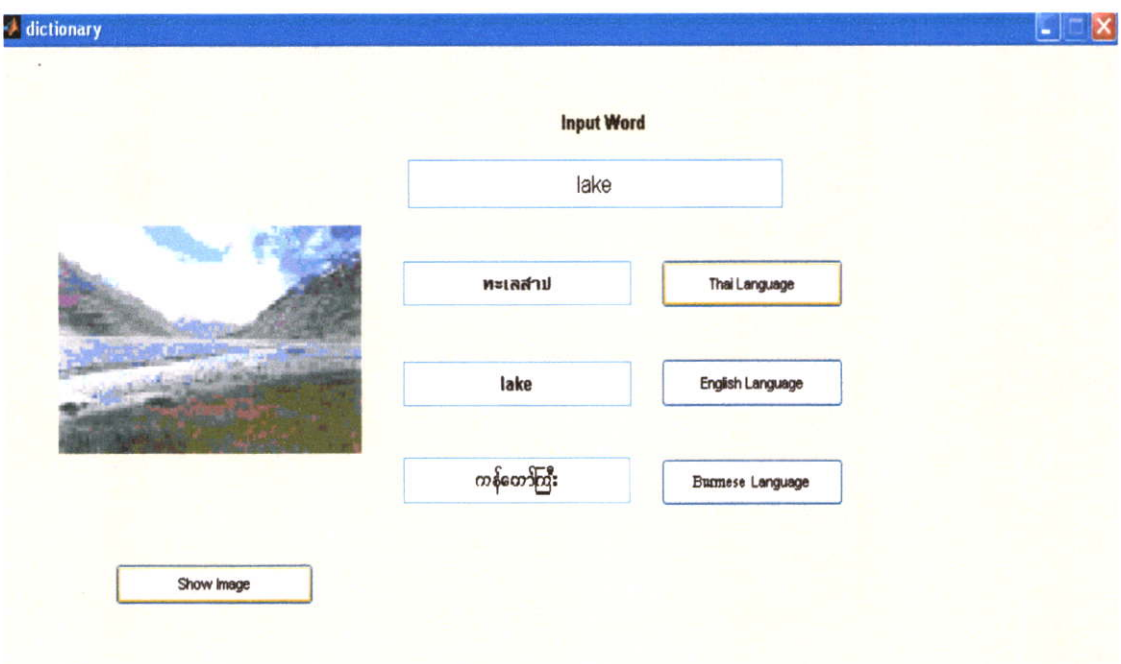
- เมื่อกำหนดอินพุตเป็นคำว่า "lake" เอาต์พุตที่ได้จะเป็นไปดังนี้

"ทะเลสาป" (Thai)

"lake" (English)

"ကန်တော်ကြီး" (Burmese)

(Picture)



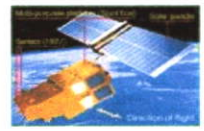
รูปที่ 5.4 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า " Lake "

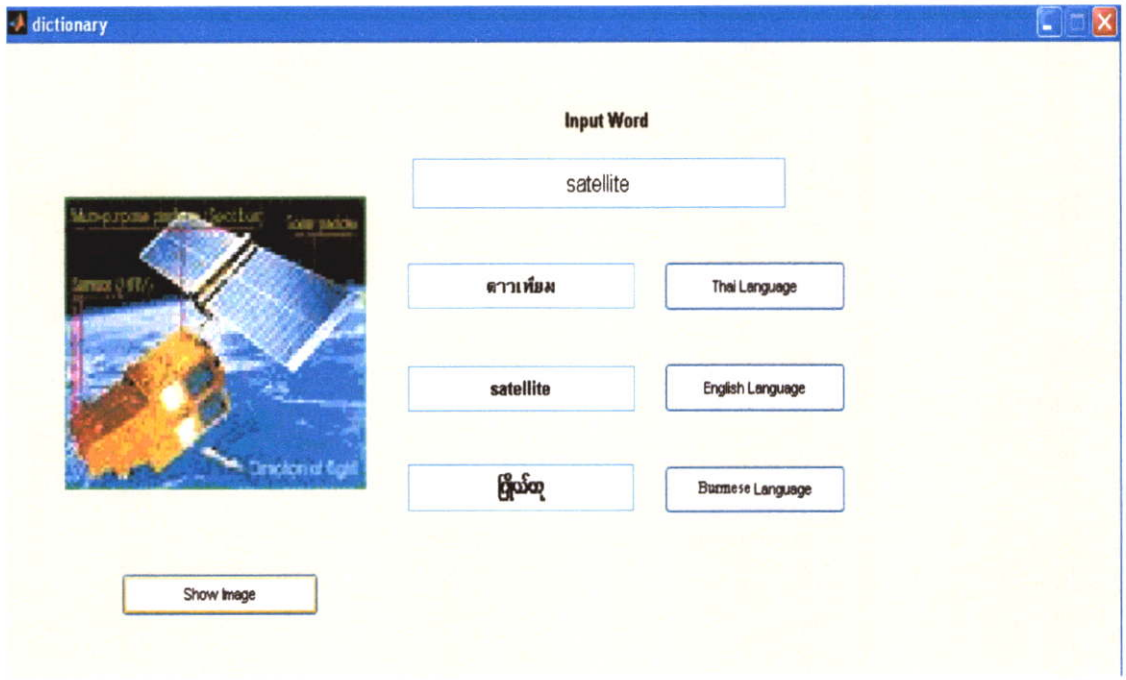
- เมื่อกำหนดอินพุตเป็นคำว่า “satellite” เอาต์พุตที่ได้จะเป็นไปดังนี้

“ดาวเทียม” (Thai)

“satellite” (English)

“ဂြိုယ်တု” (Burmese)

 (Picture)



รูปที่ 5.5 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า “ Satellite ”

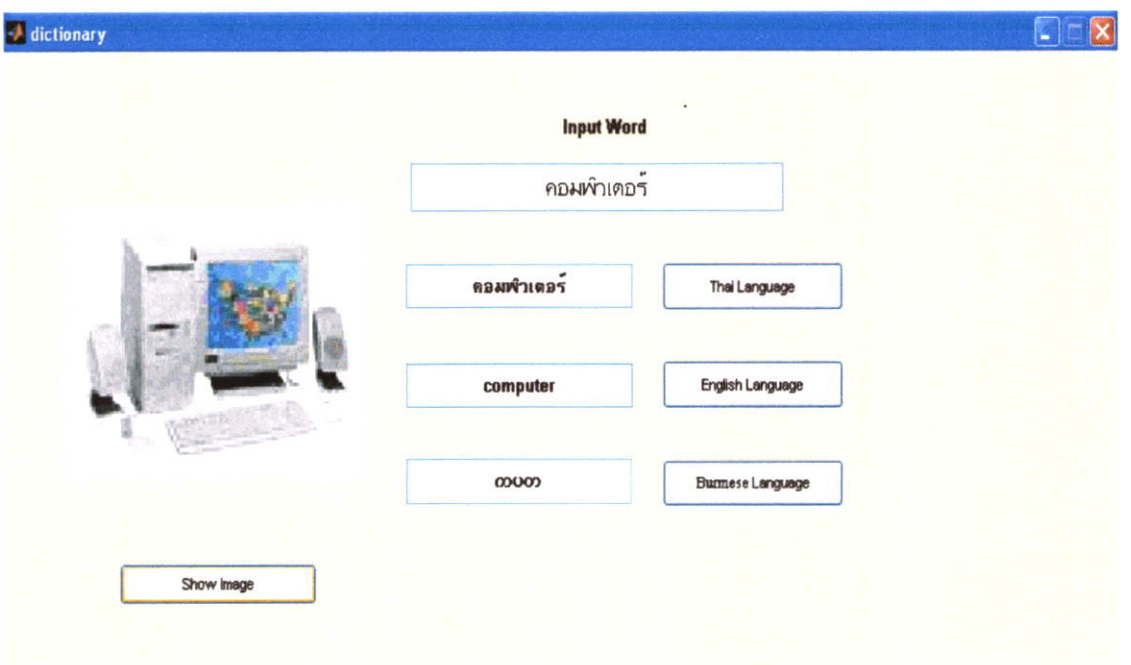
- เมื่อกำหนดอินพุตเป็นคำว่า “คอมพิวเตอร์” เอาต์พุต ที่ได้จะเป็นไปดังนี้

“คอมพิวเตอร์” (Thai)

“computer” (English)

“ ကၽတ” (Burmese)

(Picture)



รูปที่ 5.6 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “ คอมพิวเตอร์ ”

- เมื่อกำหนดอินพุตเป็นคำว่า "နကလမ" เอาต์พุตที่ได้จะเป็นไปดังนี้

"หนังสือผ่านเข้าประเทศ" (Thai)

"passport" (English)

"နကလမ" (Burmese)

(Picture)

รูปที่ 5.7 ภาพเมื่อใส่คำศัพท์เป็นภาษาพม่าคำว่า " နကလမ "

5.4 การแสดงผลเมื่อคำศัพท์นั้นไม่มีแต่จะแสดงคำศัพท์ใกล้เคียง

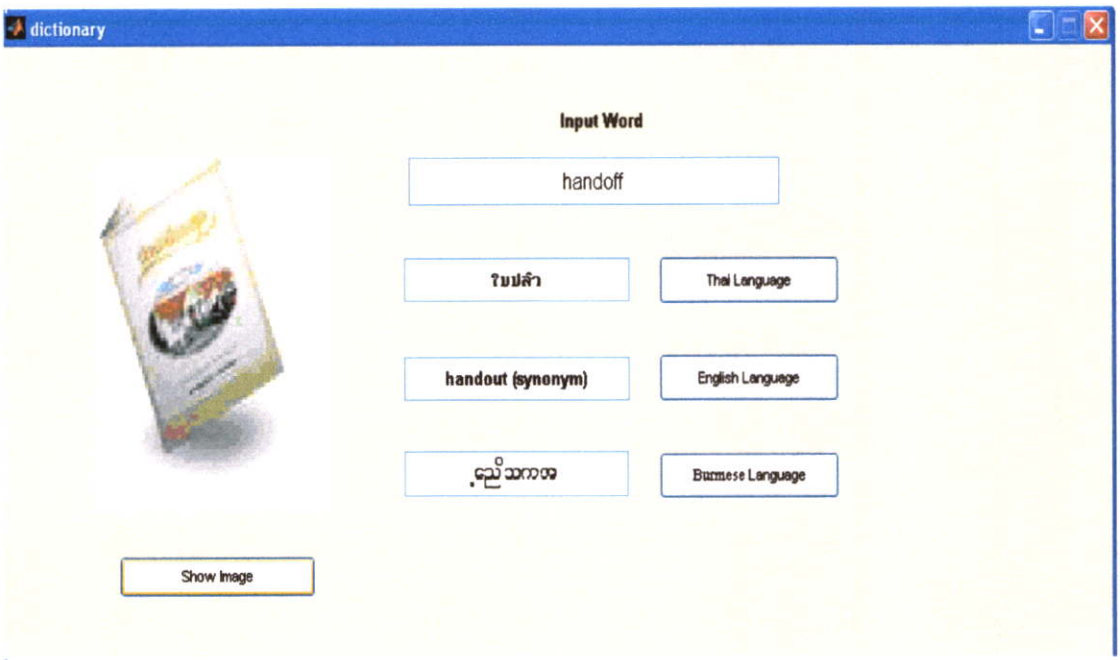
- เมื่อกำหนดอินพุตเป็นคำว่า “handoff” เอาต์พุต ที่ได้จะเป็นไปดังนี้

“ใบปลิว” (Thai)

“handout” (English)

“ ့ညှိသကာအ ” (Burmese)

(Picture)



รูปที่ 5.8 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า “ handoff ”

จากรูปที่ 5.8 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า “ handoff ” แต่ภาษาอังกฤษคำว่า “ handoff ”ไม่มี จึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือ คำว่า “ handout ” และ “handrail” เมื่อผู้ใช้เลือกคำว่า“ handout ” จะแสดงคำว่า“ handout ” ออกมาซึ่งหมายถึง ใบปลิว

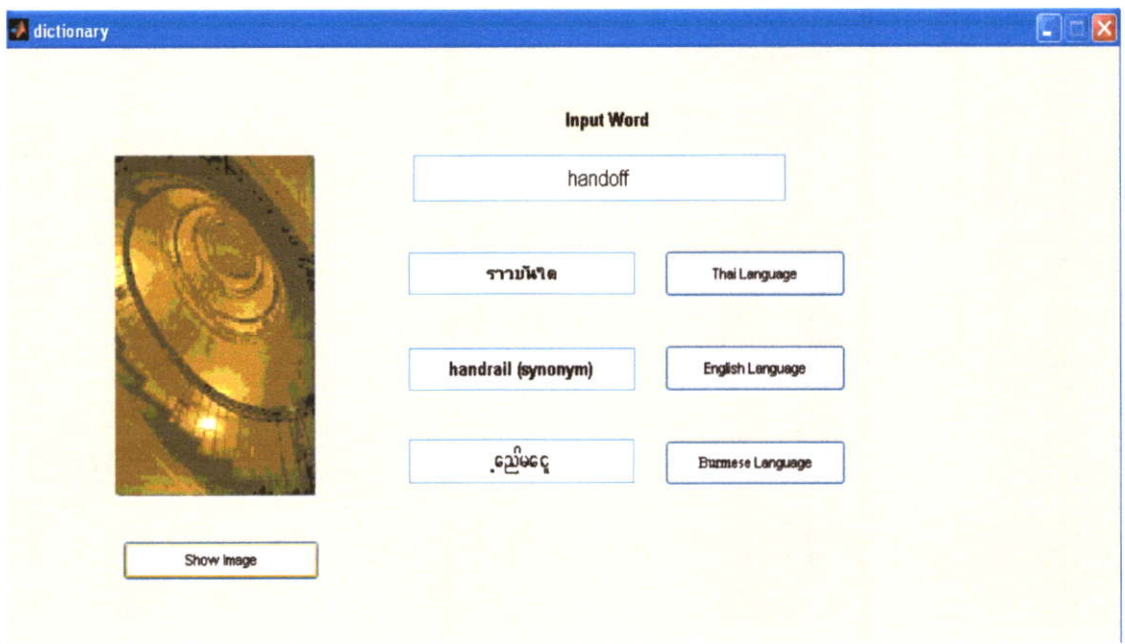
- เมื่อกำหนดอินพุตเป็นคำว่า “handoff” เอาต์พุต ที่ได้จะเป็นไปดังนี้

“ราวบันได” (Thai)

. “handrail” (English)

“ .ကြိမ်ခင်္ဂုံ” (Burmese)

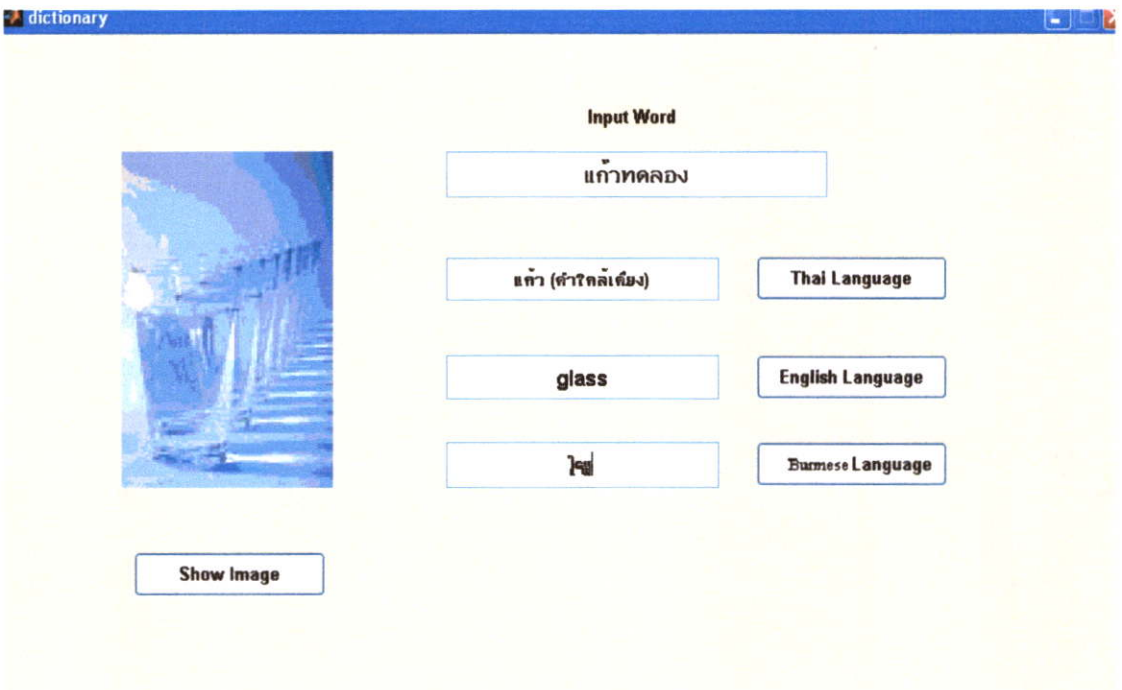
(Picture)



รูปที่ 5.9 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า “ handoff ”

จาก รูปที่ 5.19 เมื่อใส่คำศัพท์เป็นภาษาอังกฤษ คำว่า “ handoff ” แต่ภาษาอังกฤษคำว่า handoff ไม่มี จึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือ คำว่า “ handout ” และ “handrail” เมื่อผู้ใช้เลือกคำว่า “handrail”จะแสดงคำว่า “handrail” ออกมาซึ่งหมายถึง ราวบันได

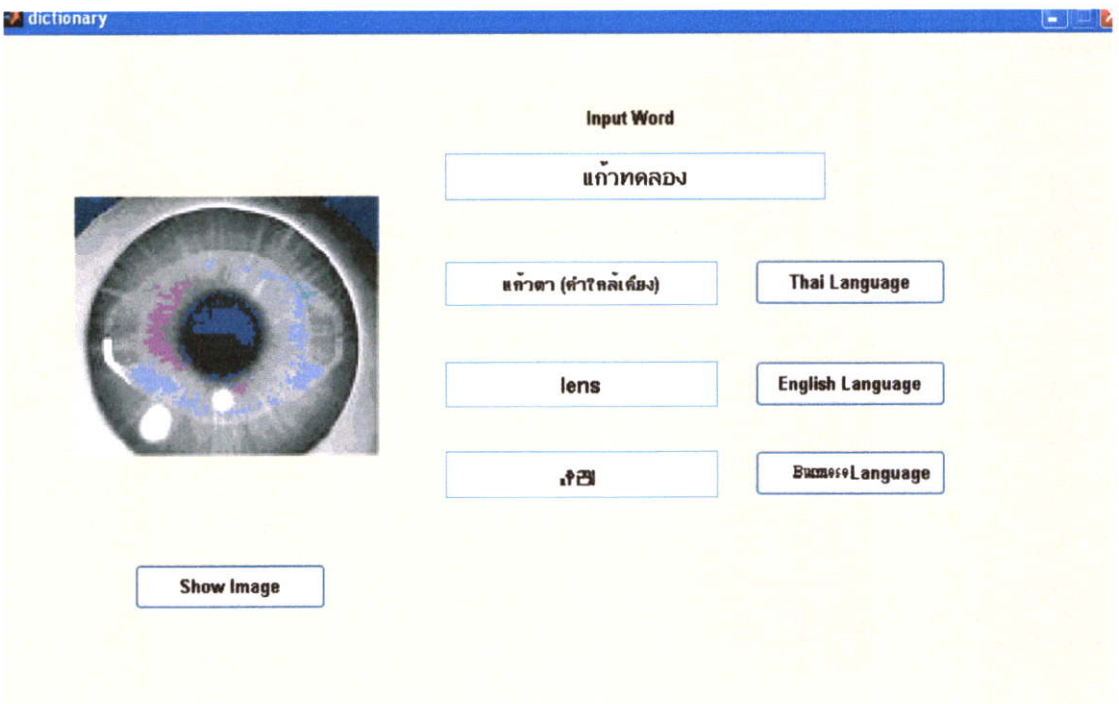
- เมื่อกำหนดอินพุตเป็นคำว่า “แก้วทดลอง” เอาต์พุตที่ได้จะเป็นไปดังนี้



รูปที่ 5.10 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “แก้วทดลอง ”

จากรูปที่ 5.10 เมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “แก้วทดลอง” ภาษาไทยคำว่า “แก้วทดลอง” ไม่มี จึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือ คำว่า “แก้ว” “แก้วตา” และ “แก้ม” เมื่อผู้ใช้เลือกคำว่า “แก้ว” จะแสดงคำว่า “ แก้ว” ออกมา

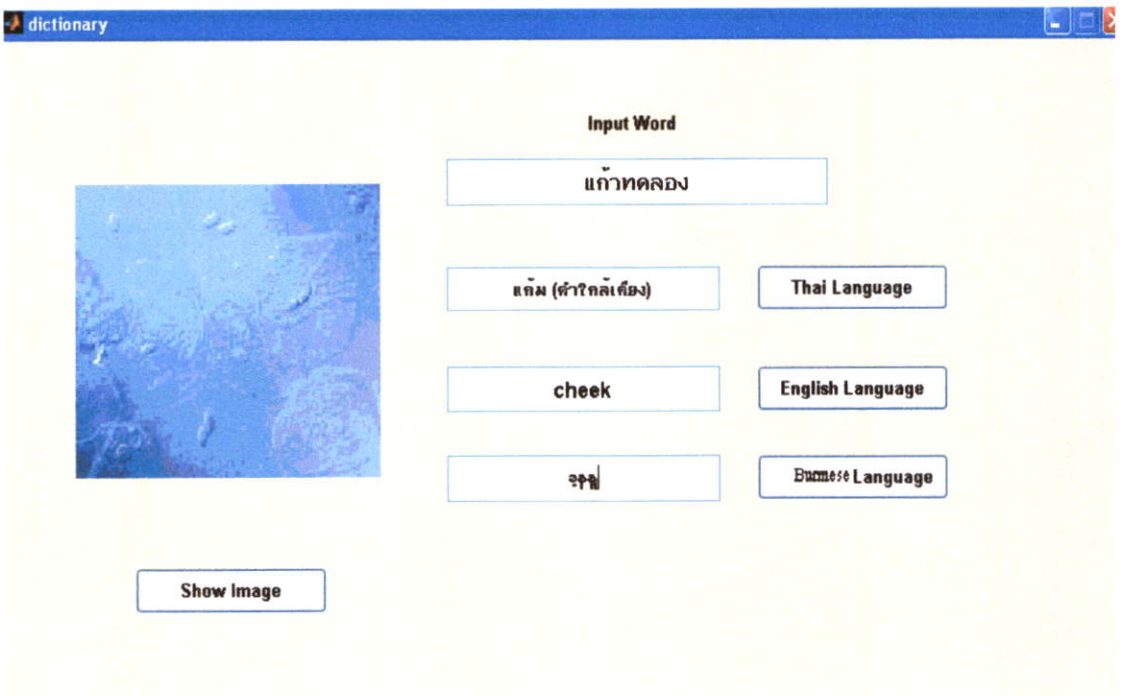
- เมื่อกำหนดอินพุตเป็นคำว่า “แก้วทดลอง” เอาต์พุตที่ได้จะเป็นไปดังนี้



รูปที่ 5.11 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “แก้วทดลอง ”

จากรูปที่ 5.11 เมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “แก้วทดลอง” แต่ภาษาไทยคำว่า “แก้วทดลอง” ไม่มี จึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือ คำว่า “แก้ว” “แก้วตา” และ “แก้ม” เมื่อผู้ใช้เลือกคำว่า “แก้วตา” จะแสดงคำว่า “ แก้วตา” ออกมา

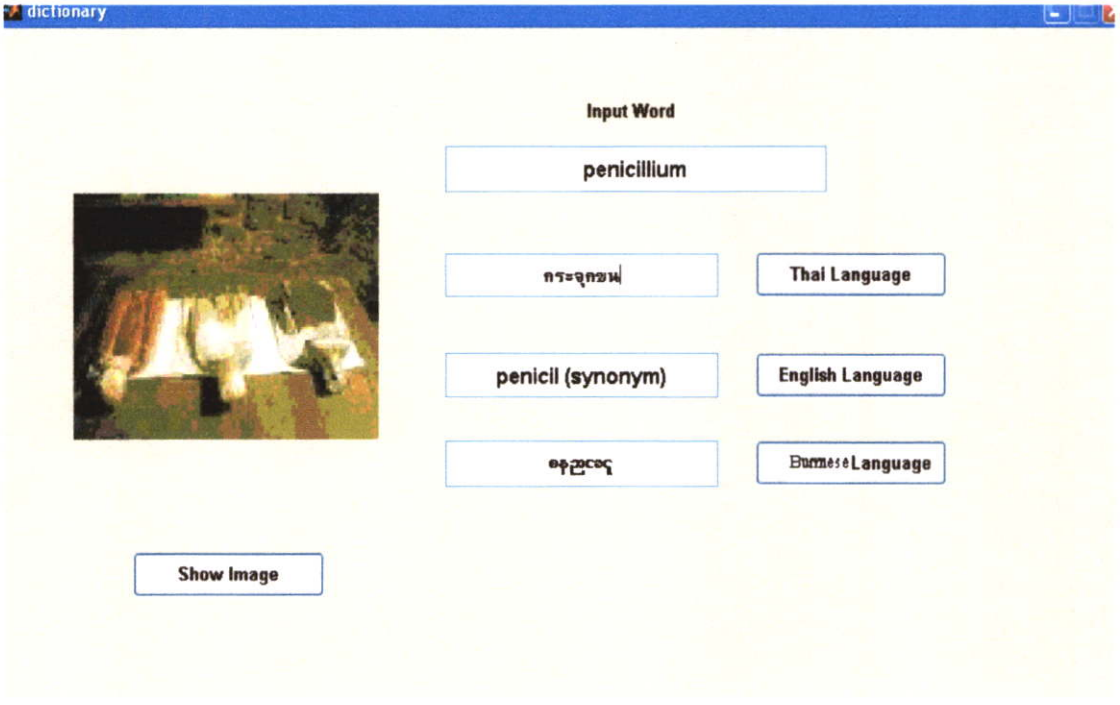
- เมื่อกำหนดอินพุตเป็นคำว่า “แก้วทดลอง” เอาต์พุตที่ได้จะเป็นไปดังนี้



รูปที่ 5.12 ภาพเมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “แก้วทดลอง ”

จากรูปที่ 5.12 เมื่อใส่คำศัพท์เป็นภาษาไทย คำว่า “แก้วทดลอง” แต่ภาษาไทยคำว่า “แก้วทดลอง” ไม่มี จึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือ คำว่า “แก้ว” “แก้วตา” และ “แก้ม” เมื่อผู้ใช้เลือกคำว่า “แก้ม” จะแสดงคำว่า“ แก้ม” ออกมา

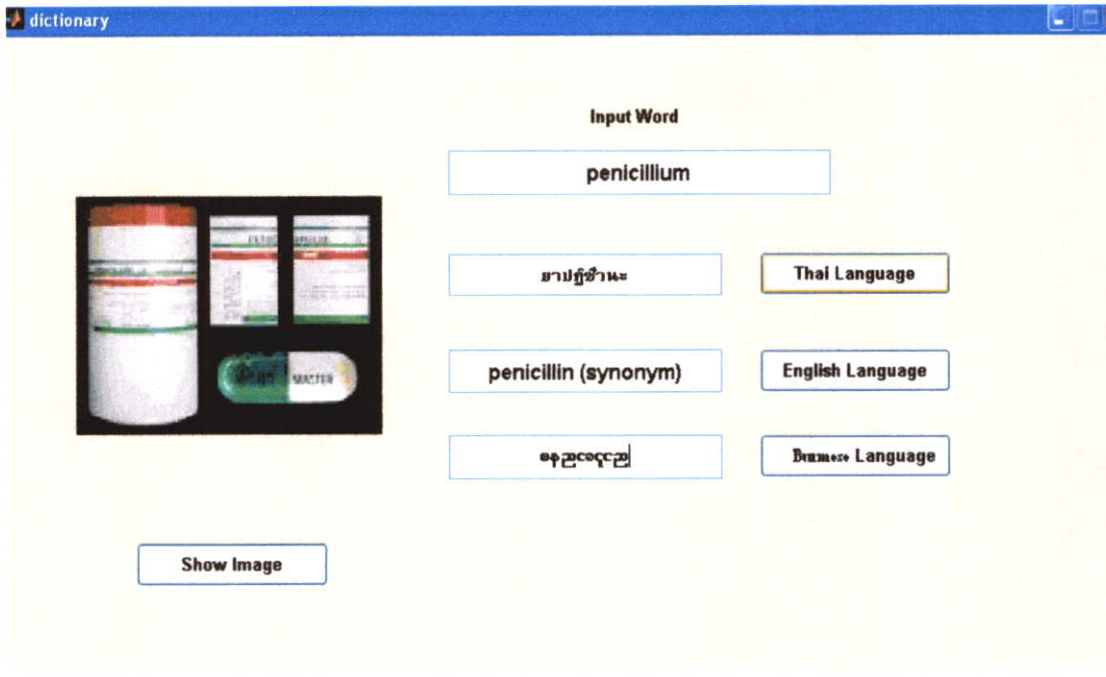
- เมื่อกำหนดอินพุตเป็นคำว่า “penicillium” เอาต์พุตที่ได้จะเป็นไปดังนี้



รูปที่ 5.13 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษคำว่า “penicillium”

จากรูปที่ 5.13 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษคำว่า “penicillium” แต่ภาษาอังกฤษคำว่า“penicillium”ไม่มี จึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือ คำว่า “penicil” และ “penicillin” เมื่อผู้ใช้เลือกคำว่า “penicil” จะแสดงคำว่า “penicil” ซึ่งหมายความว่า กระจุกขน

- เมื่อกำหนดอินพุตเป็นคำว่า “penicillium” เอาต์พุตที่ได้จะเป็นไปดังนี้

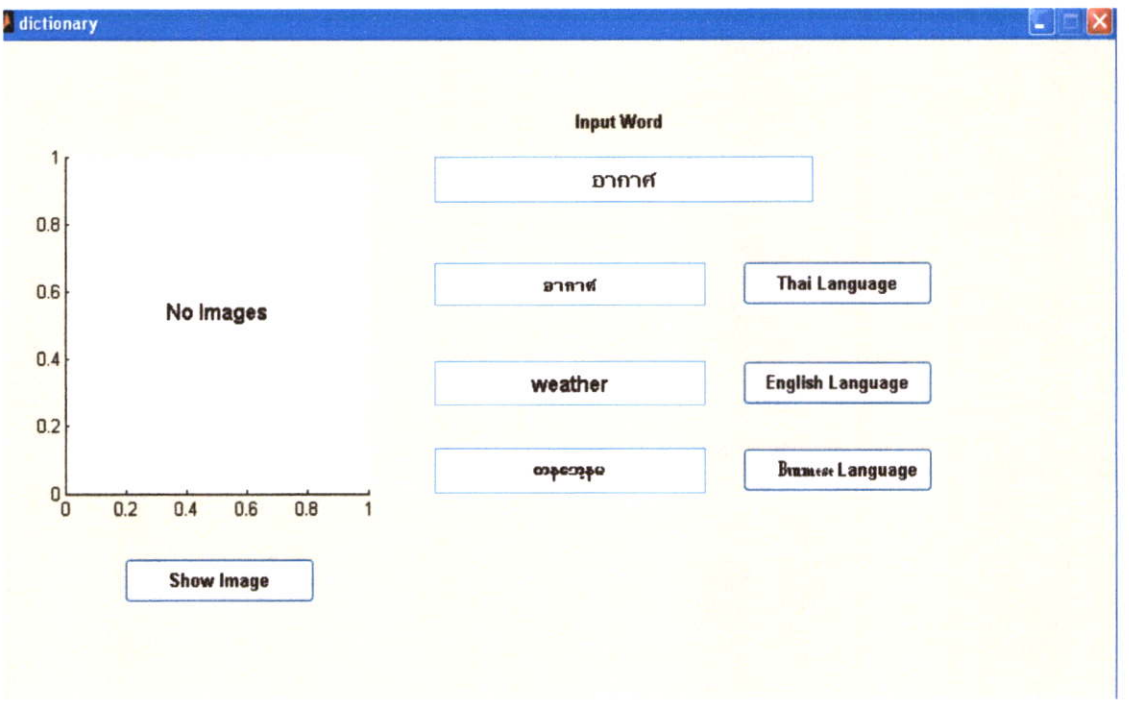


รูปที่ 5.14 ภาพเมื่อใส่คำศัพท์เป็นภาษาอังกฤษคำว่า “penicillium”

จากรูปที่ 5.14 เมื่อใส่คำศัพท์เป็นภาษาอังกฤษคำว่า “penicillium” ไม่มีจึงแสดงคำศัพท์ที่ใกล้เคียงที่สุดออกมาคือคำว่า “penicillin” และ “penicil” เมื่อผู้ใช้เลือกคำว่า “penicillin” จะแสดงคำว่า “penicillin” ซึ่ง หมายถึง ยาปฏิชีวนะ

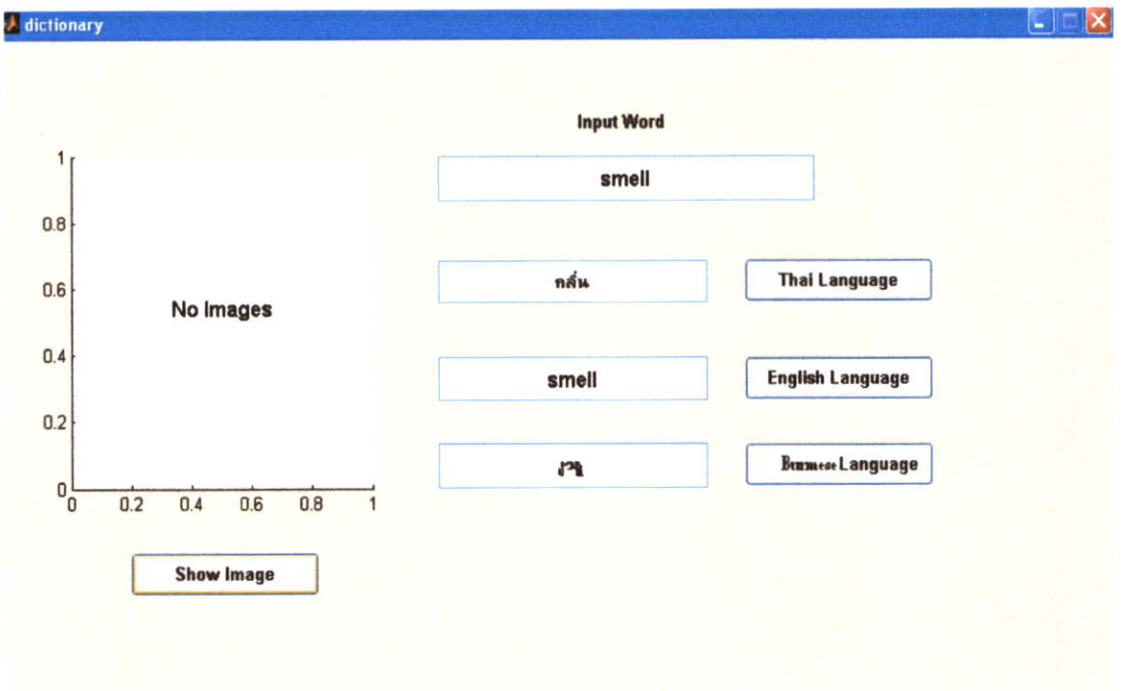
5.5 การแสดงผลเมื่อคำศัพท์นั้นไม่มีภาพ

- เมื่อกำหนดอินพุตเป็นคำว่า “อากาศ” เอาต์พุตที่ได้จะเป็นไปดังนี้



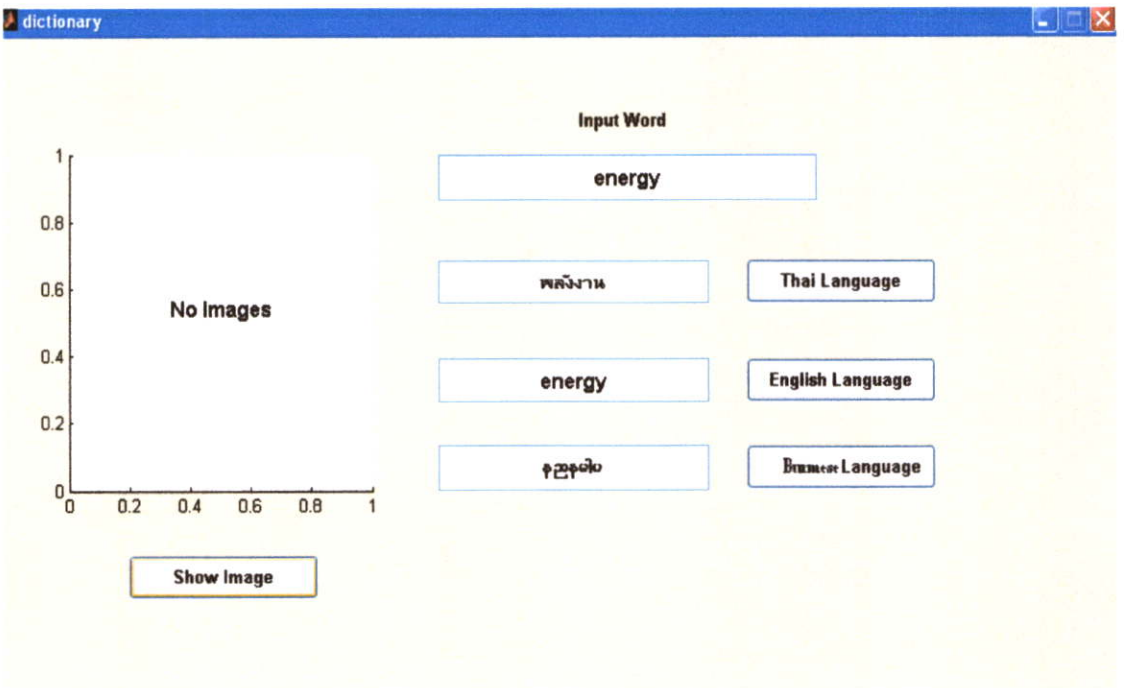
รูปที่ 5.15 อินพุต / เอาต์พุตคำว่า “อากาศ”

- เมื่อกำหนดอินพุตเป็นคำว่า “smell” เอาต์พุตที่ได้จะเป็นไปดังนี้



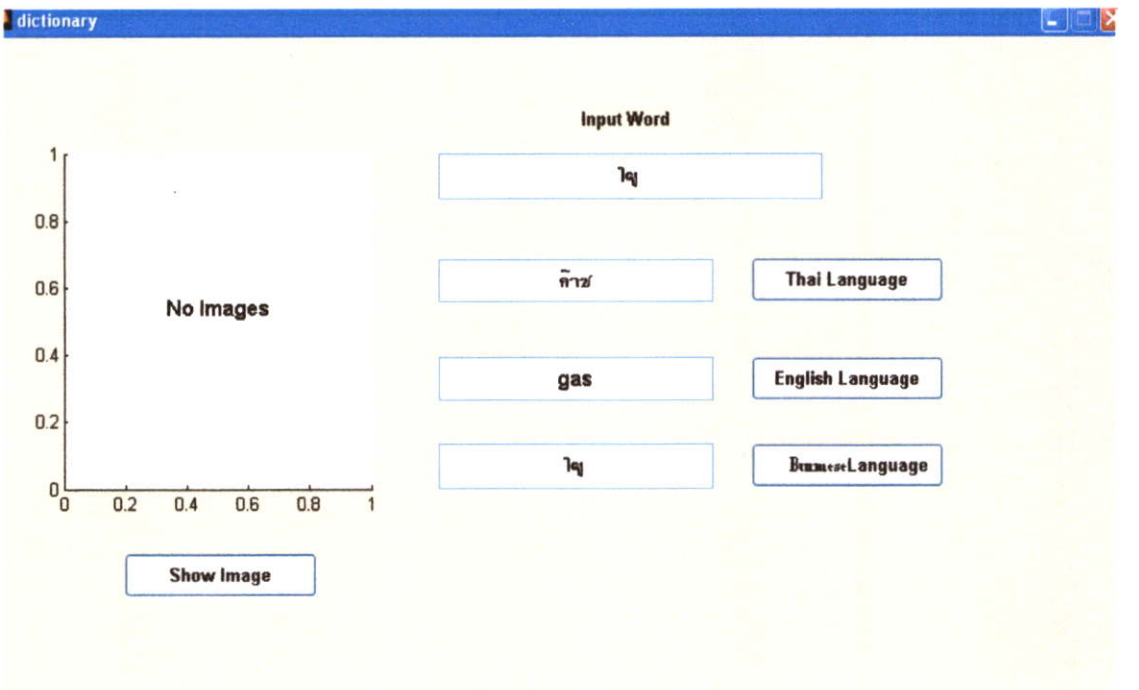
รูปที่ 5.16 อินพุต / เอาต์พุตคำว่า “smell”

- เมื่อกำหนดอินพุตเป็นคำว่า “energy” เอาต์พุตที่ได้จะเป็นไปดังนี้



รูปที่ 5.17 อินพุต / เอาต์พุตคำว่า “energy”

- เมื่อกำหนดอินพุตเป็นคำว่า "ใฌ" เอาต์พุตที่ได้จะเป็นไปดังนี้



รูปที่ 5.18 อินพุต / เอาต์พุตคำว่า "ใฌ"

5.6 ผลการทดลองของคำศัพท์อื่น ๆ

- เมื่อกำหนดอินพุตเป็นคำว่า **"sea"** เอาต์พุตที่ได้จะเป็นไปดังนี้

"ทะเล" (Thai)

"marine; sea" (English)

" ပင်လယ " (Burmese)

(Picture)

- เมื่อกำหนดอินพุตเป็นคำว่า **"ดาวเสาร์"** เอาต์พุตที่ได้จะเป็นไปดังนี้

"ดาวเสาร์" (Thai)

"saturn" (English)

"စနေဂြိဟ" (Burmese)

(Picture)

- เมื่อกำหนดอินพุตเป็นคำว่า " ခြင္ကူ " เอาต์พุตที่ได้จะเป็นไปดังนี้

"ชอล์ค" (Thai)

"crayon; pastel" (English)

"ခြင္ကူ" (Burmese)

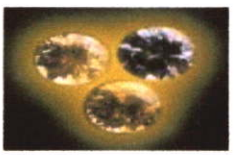
(Picture)

- เมื่อกำหนดอินพุตเป็นคำว่า "**เพชร**" เอาต์พุตที่ได้จะเป็นไปดังนี้

"เพชร" (Thai)

"diamond" (English)

"စိန်ကျောက်" (Burmese)



(Picture)

บทที่ 6

สรุปผลการวิจัย

สำหรับในวิทยานิพนธ์ฉบับนี้ ได้นำเสนอพจนานุกรม ภาษาไทย ภาษาอังกฤษ และภาษาพม่า โดยใช้หลักการของการค้นหาข้อมูล แบบไบนารีเซิร์ททรี (Binary search trees) เนื่องจากวิธีของไบนารีเซิร์ททรีนี้ เป็นวิธีที่มีความเร็วในการค้นหาข้อมูลและยังเหมาะสมกับข้อมูลที่มีขนาดใหญ่ด้วย ส่วนของอินพุตสามารถป้อนคำศัพท์ที่ต้องการค้นหาความหมายภาษาใดภาษาหนึ่ง ตัวอย่างเช่น ภาษาไทย หรือ ภาษาอังกฤษ หรือ ภาษาพม่า จากนั้นโปรแกรมจะทำการประมวลผลเพื่อค้นหาความหมายของคำศัพท์ตามที่ต้องการแล้วทำการแสดงผลออกมาให้เห็นความหมายของคำศัพท์ทั้งสามภาษา ไม่ว่าจะค้นหาคำศัพท์ด้วยภาษาใดก็ตาม โดยในส่วนของเอาต์พุตสามารถแสดงภาษาได้หลายภาษาได้แก่ ภาษาไทย อังกฤษ พม่า และนอกจากจะแสดงภาษาได้หลายภาษาแล้ว ยังสามารถแสดงภาพ และ เสียงประกอบในเวลาเดียวกันด้วย แต่ในกรณีที่คำศัพท์ที่ต้องการค้นหาไม่มีอยู่ในฐานข้อมูล โปรแกรมจะประมวลผลโดยแสดง ผลลัพธ์ เป็นคำศัพท์ใกล้เคียงขึ้นมาให้ผู้ใช้งานทำการเลือกความหมายของคำศัพท์ที่ใกล้เคียงหรือต้องการ

จากผลการทดสอบ โดยการสุ่มตัวอย่างคำศัพท์จำนวน 500 คำ และนำเวลาที่ใช้ในการค้นหาคำศัพท์เพียง 25 คำ มาพิจารณาจากจำนวนคำศัพท์ที่แตกต่างกันได้ผลดังต่อไปนี้ จากผลการทดสอบในตารางที่ 5.1, 5.2 และ 5.3 ในบทที่ 5 สรุปได้ว่าการค้นหาคำศัพท์ด้วยวิธีนอมอลไลซ์ไบนารีเซิร์ททรีจะใช้เวลาค้นหาคำศัพท์เฉลี่ย 0.607, 0.835 และ 1.477 วินาที จากคำศัพท์ทั้งหมด 10,000 15,000 และ 20,000 คำ ตามลำดับ เมื่อเปรียบเทียบกับการค้นหาคำศัพท์ด้วยวิธี N-best[8] จะต้องใช้เวลาในการค้นหาคำศัพท์เฉลี่ย 2.908, 3.581 และ 5.163 วินาที ตามลำดับ และวิธี Lattice[12] ใช้เวลาเฉลี่ย 5.29, 6.358 และ 7.296 วินาที ตามลำดับ จะเห็นได้ว่าวิธีนอมอลไลซ์ไบนารีเซิร์ททรีสามารถค้นหาคำศัพท์ได้เร็วกว่าการค้นหาคำศัพท์ด้วยวิธี N-best[8] และ Lattice[12]

จากผลการทดสอบในรูปที่ 5.1, 5.2 และ 5.3 สรุปได้ว่าการค้นหาคำศัพท์โดยวิธีที่นำเสนอโดยการสุ่มคำศัพท์จำนวน 500 คำ จากฐานข้อมูลคำศัพท์ 10,000, 15,000 และ 20,000 คำ จะใช้เวลาค้นหาเฉลี่ย 0.725, 0.963 และ 1.586 วินาที ตามลำดับ ซึ่งสามารถค้นหาคำศัพท์ได้เร็วกว่าวิธีเดิม

นอกจากนี้วิธีที่นำเสนอในวิทยานิพนธ์นี้ยังมีความถูกต้อง แม่นยำ ในการค้นหาคำศัพท์อีกด้วย สำหรับในอนาคตสิ่งที่ควรจะพัฒนาต่อไป คือ การออกแบบอัลกอริทึมที่ใช้ในการค้นหาข้อมูล ซึ่งอาจจะมีการนำเอาวิธีอื่นมาช่วยนอกเหนือจากวิธีไบนารีเซิร์ททรี (Binary search tree)

บรรณานุกรม

[1] L. Nguyen, R. Schwarts, Y. Zhao and G. Zavaliagkos, "Is N-best dead?," Proc. of the ARPA human language technology workshop, March 1994.

[2] M. Rayner, D. Carter, V. Digalakis and P. Price, "Combining knowledge sources to recorder N-best speech hypothesis lists," Proc. of the ARPA human language technology workshop, March 1994.

[3] M. Ostendorf, A. Kannan, S. Austin, O. Kimball, R. Schwartz and J.R. Rohlicek, "Integration of diverse recognition Methodologies through reevaluation of N-best sentence hypotheses," Proc. of the DARPA Workshop on speech and natural language, pp. 83-87, Feb. 1991.

[4] F. Alleva, X. Huang and M. Hwang, "An improved search algorithm using incremental knowledge for continuous speech recognition," IEEE Int. conf. Acoust., Speech, Signal processing, pp. 307-310, Apr. 1993.

[5] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, "Large-Vocabulary dictation using SRI's Decipher tm speech recognition system: progressive search technique," IEEE Int. conf. Acoust., Speech, Signal Processing, pp. 319-322, Apr. 1993.

[6] M. Ostendorf, F. Richardson, S. Tibrewal, R. Iyer, O. Kimball and J.R. Rohlicek, "Stochastic segment modeling for CSR: The BU WSJ benchmark system," ARPA Spoken Language Systems Technology Workshop, pp. 255-259 March 1994.

[7] G. Zavaliagkos, Y. Zhao, R. Schwartz and J. Makhoul, "A hybrid segmental neural net/hidden markov model system for continuous speech recognition," IEEE Trans. on speech and audio processing, Vol. 2, No. 1, pp. 151-160, March 1994.

[8] A. Kannan, M. Ostendorf and J.R Rohlicek, "Weight estimation for N-best rescoring," Proc. of the DAARPA Workshop on speech and natural language, pp. 455-456, Feb. 1992.

[9] V. Digalakis, M. Ostendorf and J.R. Rohlicek, "Fast search algorithms for phone classification and recognition using segment-based model," IEEE Trans. on signal processing, pp. 2885-2896, Dec. 1992.

[10] F. Kubala, et al., "The hub and spoke paradigm for CSR Evaluation," Proc. of the ARPA Human language technology workshop, pp. 354-359, March 1994.

[11] R. Iyer, M. Ostendorf and J.R. Rohlicek, "Language modeling with sentence-level mixtures," Proc. of the ARPA workshop on Human language technology, pp. 951-955, March 1994.

[12] F. Richardson, M. Ostendrof and J.R. Rohliek "Lattice-base search strategies for large vocabulary speech recognition," IEEE Con. ICASSP-95, pp. 576-579, May 1995.

[13] M. A. Weiss., Data Structures and Algorithm Analysis, Benjamin/Cummings Publishing Company, Inc. 1995.

[14] S. Lipschutz, Data Structure, McGraw-Hill, Inc. 1997.

[15] ทรงลักษณ์ พิริยะไพโรจน์ "เรียนลัด Data Structure ด้วย Visual Basic" กรุงเทพฯ: โปรวิชั่น 2544

[16] รวิวรรณ เทนอิสสระ "ฐานข้อมูลและการออกแบบ" กรุงเทพฯ: เธิร์ดเวฟ เอ็ดดูเคชั่น จำกัด 2543

[17] อำไพ พรประเสริฐกุล "การวิเคราะห์และออกแบบระบบ" กรุงเทพฯ: ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ 2543

[18] สุชาย ธนวเสถียร, วิชัย จิวังกูร "โครงสร้างข้อมูลเพื่อการออกแบบโปรแกรมคอมพิวเตอร์" กรุงเทพฯ: ซีเอ็ดยูเคชัน จำกัด 2530

ภาคผนวก

ภาคผนวก ก

"Pseudocode for Searching"

```
functionsearch (x : integer; T : search trees)
  if x<T
      search := search(x, T.left)
  else x>T
   search := search(x, T.right)
end.

Assign  :x1 = ('English Languages')
    x2 = ('Thai Languages')
    x3 = ('Myanmar Languages')

NTE = Input ('Thai or English or Myanmar Languages')

if NTE =    x1
    search (x1,T)

  fori = 1: Data_Base
   compare (x1 , Data_Base(i)
  if x1  =  1
   display (Data_Base(i))
 else
     display (No Data Information)
  end
end
          elseif NTE =    x2
   search (x2,T)
```

```
    for i = 1: Data_Base
       compare (x2 , Data_Base(i)
      if x2  =  1
       display (Data_Base(i))
    else
     display (No Data Information)
   end
end

      else NTE =    x3
 search (x3,T)

    fori = 1: Data_Base
         compare (x3 , Data_Base(i)
      if x3  =  1
       display (Data_Base(i))
  else
 display (No Data Information)
    end
   end
 en
```

โปรแกรม

```
function varargout = dictionary(varargin)
% DICTIONARY M-file for dictionary.fig
%      DICTIONARY, by itself, creates a new DICTIONARY or raises the
existing
%      singleton*.
%
%      H = DICTIONARY returns the handle to a new DICTIONARY or the
handle to
%      the existing singleton*.
%
%      DICTIONARY('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in DICTIONARY.M with the given input
arguments.
%
%      DICTIONARY('Property','Value',...) creates a new DICTIONARY or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before dictionary_OpeningFunction gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to dictionary_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help dictionary

% Last Modified by GUIDE v2.5 24-Mar-2007 22:12:26

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @dictionary_OpeningFcn, ...
                   'gui_OutputFcn',  @dictionary_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end


if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
% End initialization code - DO NOT EDIT
% --- Executes just before dictionary is made visible.

function dictionary_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to dictionary (see VARARGIN)

% Choose default command line output for dictionary
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes dictionary wait for user response (see UIRESUME)
% uiwait(handles.figure1);

[BT,AT] = xlsread('Book313.xls');
n=max(size(AT));
m=max(size(NST));
x=char(AT(:,1));
x1=char(AT(:,2));

y=char(AT(:,3));
y1=char(AT(:,4));

z=char(AT(:,5));

        for i=1:n

                A = strcmp(NST,x(i,1:m));
                Q(i)=A;
                    if A==1
                         nameThai=x(i,:);
                         nameEnglish=y(i,:);
                         nameMyanmar=z(i,:);
                         disp([nameEnglish])
                         disp([nameThai])
                         disp([nameMyanmar])
                         fprintf('\n')
                         fprintf('\n')
                    end

                 B = strcmp(NST,y(i,1:m));
                 P(i)=B;
                    if B==1
                         nameThai=x(i,:);
                         nameEnglish=y(i,:);
                         nameMyanmar=z(i,:);
                         disp([nameEnglish])
                         disp([nameThai])
                         disp([nameMyanmar])
                         fprintf('\n')
```

```
                    fprintf('\n')

                end
            C = strcmp(NST,z(i,1:m));
            K(i)=C;
                if C==1
                    nameThai=x(i,:);
                    nameEnglish=y(i,:);
                    nameMyanmar=z(i,:);
                    disp([nameEnglish])
                    disp([nameThai])
                    disp([nameMyanmar])
                    fprintf('\n')
                    fprintf('\n')
                end
        end

j=1:n;
q=sum(Q);
p=sum(P);
k=sum(K);

    if q==1
            q1=j(Q);
            t=imread( [num2str(q1),'.jpg'] );
            image(t)
            [h,fs]=wavread([num2str(q1),'.wav']);
            sound(h,fs)
            break
    end

    if p==1
            p1=j(P);
            t=imread( [num2str(p1),'.jpg'] );
            image(t)
            [h,fs]=wavread([num2str(p1),'.wav']);
            sound(h,fs)
            break
    end

    if k==1
            k1=j(K);
            t=imread( [num2str(k1),'.jpg'] );
            image(t)
            [h,fs]=wavread([num2str(k1),'.wav']);
            sound(h,fs)
            break
    end

anyname=input('insert anything in Thai or English or Myanmar  = ');
M=max(size(anyname));
for i=1:M

    D = strcmp(sername,x1(i,1:M));
       if D==1
            t=imread( [num2str(M),'.jpg'] );
            image(t)
```

```
            [h,fs]=wavread([num2str(M),'.wav']);
            sound(h,fs)
        end

    E = strcmp(sername,y1(i,1:M));
        if E==1
            t=imread( [num2str(M),'.jpg'] );
            image(t)
            [h,fs]=wavread([num2str(M),'.wav']);
            sound(h,fs)
        end

end

guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = dictionary_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in thai.
function thai_Callback(hObject, eventdata, handles)
% hObject    handle to thai (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.thai = get(handles.edit1,'value')
set(handles.edit1,'string',str2num(handles.thai));

% --- Executes on button press in english.
function english_Callback(hObject, eventdata, handles)
% hObject    handle to english (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.english = get(handles.edit2,'value')
set(handles.edit2,'string',str2num(handles.english));

% --- Executes on button press in myanmar.
function myanmar_Callback(hObject, eventdata, handles)
% hObject    handle to myanmar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.myanmar = get(handles.edit3,'value')
set(handles.edit3,'string',str2num(handles.myanmar));
```

```
% --- Executes on button press in show_image.
function show_image_Callback(hObject, eventdata, handles)
% hObject    handle to show_image (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
imshow('handrail.jpg')


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called


% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4
as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

ภาคผนวก ข

ตัวอย่างฐานข้อมูล 3 ภาษา

(สำหรับข้อมูลภาพและเสียงที่เก็บอยู่ในรูปแฟ้มข้อมูลภาพและเสียงไม่ได้นำมาแสดงในนี้ด้วย)

	Thai_Language	English-Language	Burmese_language
1	จักรวาล	cosmic	စွတ္တဝါ
2	ดาว	star	ကြယ်
3	ดาวตก	shooting star	ကယ်ပုံ
4	ดาวหาง	comet	ကယ်ဝံစုံ
5	ดาวประกายพฤกษ์	morning star	ကယ်စု
6	ดาวเคราะห์	planets in the solar system	ဂြိုဟ်
7	ดาวอังคาร	Mars	ဒဂ္ဂါဂြိုဟ်
8	ดาวพุธ	Mercury	ဗုဒ္ဓဟူးဂြိုဟ်
9	ดาวพฤหัสบดี	Jupiter	ကြာသပတေးဂြိုဟ
10	ดาวศุกร์	Hesperus	သောကြာဂြိုဟ်
11	ดาวเสาร์	Saturn	စနေဂြိုဟ်
12	วันอาทิตย์	Sunday	တနင်္ဂနွေဂြိုဟ်
13	พระอาทิตย์	sun	နေ
14	โลก	earth	ထွာဒေ့မြေ
15	แสง	colour	အရောင်
16	แสงเดือน	moonlight	လရောင်
17	แสงอาทิตย์	sunshine	နေရောင်
18	พายุ	strom	မုန်တိုင်း

19	ฤดูฝน	rainy season	မိုး
20	ฝนตก	rain	မိုးရွာသည်
21	ท้องฟ้า	sky	ကောင်းကင် ။ မိုး
22	ฟ้าคะนอง , ฟ้าร้อง	thunder	မိုးချုန်းသည်
23	ฟ้าผ่า	thunderclap	မိုးကြိုးပစ်သည်
24	ฟ้าแลบ	lightning	မိုးကြိုးထစ်သည်
25	บรรยากาศ	atmosphere	လေထု။လေလွှာ
26	ดิน	ground	မြေ ။ ပထဝီ
27	น้ำ	water	ရေ ။ အပေါ
28	ลม , อากาศ	air	လေ ။ ဝါယော
29	ไฟ	fire	မီး ။ တေဇော
30	เมฆ	cloud	မိုးတိမ်
31	ความเย็น	cold	အအေး
32	อ่าว	bay	ပင်လယ်အော်
33	ฝั่งทะเล , ชายฝั่ง	seaboard	ပင်လယ်ကမ်း
34	โคลน , ตม	mud	နွံ
35	คลอง	canal	တူးမြောင်း
36	แหลม	cape,headland	အဂူ ရေတံခွန်
37	ถ้ำ	cave	ဂူ
38	เหว	abyss	ချောက် ။ ကျောက်

39	หน้าผา	cliff	ကမ်းစောက်
40	ห้วย	bog	ချောင်းကလေး
41	สายน้ำ , กระแสน้ำ	current	ရေစီး ။ ဒီရေ
42	น้ำตก	waterfall(s); falling water; (a) fall(s)	ရေတံခွန်
43	น้ำขึ้น	rising tide	ရေတက်
44	ป่า	forest	တော
45	ภูเขา	mountain	တောင်
46	เนิน , โคก	hill	တောင်ကုန်း
47	ยอดเขา	mountain peak	တောင်ထိပ်
48	น้ำแข็ง	ice	ရေခဲ ။ ရေခဲရည်
49	เกาะ	island	ကျွန်း
50	ทะเลสาป	influent	ရေဘိုင်ကြီး
51	บก	bend	ကုန်း
52	ท้องทุ่ง	bjin	ကွင်းပြင်
53	เทือกเขา	range of hills	တောင်တန်း
54	หิน	rock	ကျောက်
55	กรวด	pebble	ကျောက်စရစ်
56	ทราย	sand	သဲ
57	หาด	sand bank	သောင်
58	ทะเล	sea	ပင်လယ်

59	มหาสมุทร	ocean	ဥမ္မဂ္ဂာ
60	น้ำพุ	unboiled water	ရေစမ်း
61	พายุ	storm	မုန်တိုင်း
62	ลำธาร , ทางน้ำไหล	lie in wait for the enemy	ချောင်းကလေး
63	น้ำจืด	fresh water	ရေချို
64	น้ำเค็ม	salt-water	ရေငန်
65	อ่างเก็บน้ำ	poll	ရေကန်
66	คลื่น	wave	လှိုင်း
67	บ่อ	well	ရေတွင်း
68	น้ำวน	maelstrom	ရေဝဲ
69	สารส้ม	alum	ကျောက်ချဉ်
70	อำพัน	amber	ပယင်း
71	พลวง	antimony	ခနောက်စိမ်း
72	สารหนู	arsenic	စိန်
73	น้ำประสาน	borax	လက်ချား
74	ทองเหลือง	brass	ဖွေးဝါ
75	อิฐ	firebrick	အုတ်
76	ทองบรอนซ์ , ทองแดง	copper	ဖွေးနီ
77	ปูนซีเมนต์	cement	ဘိလပ်မြေ
78	ชอล์ค	crayon; pastel	မြေဖြူခဲ

79	ดินเหนียว	terra cotta	မြေစေး
80	ถ่านหิน	anthracite	ကျောက်မီးသွေး
81	หินปะการัง	coral ,polyp	ကျောက်ပန်း ၊သွာ
82	ผลึก	crystal	ကျောက်သလင်း
83	เพชร	diamond	စိန်ကျောက်
84	มรกต	emerald	မြ
85	ทอง	gold	ရွှေ
86	เหล็ก	alloy steel	သံ
87	ตะกั่ว	plumbum	ခဲ
88	หินปูน , ปูนขาว	lime	ထုံးဖု
89	หินอ่อน	marble	ကျောက်ဖူ
90	ปรอท	mercury	ပြဒါး
91	มุกดา	girasol	မဟူရာ
92	สินแร่	metal	သတ္တု
93	มุก	pearl	ပုလဲ
94	ปิโตรเลียม	petroleum	ရေနံ
95	ทับทิม	ruby	ကျောက်နီ
96	เกลือ	salt	ဆား
97	นิล	sapphire	နီလာ
98	เงิน	silver	ငွေ

99	โซดา	soda	ပြာဆား
100	เหล็กกล้า	alloy steel	သံမို
101	กำมะถัน	sulphur	ကန်
102	เงินบริสุทธิ์	silver of the finest quality	ဘော်
103	ดีบุก	tin	
104	สังกะสี	zinc	သွတ်
105	สัตว์	animal	သား
106	อีเก้ง	Syn	ကျီ
107	หมี	bear	ဝက်ဝံ
108	นก	bird	ငှက်
109	ควาย	buffalo	ကွဲ
110	วัว	bull,cow	နွား
111	แมว	cat	ဓွကာင်
112	ไก่	chicken	ွကက်
113	ม้า	horse	မြင်း
114	ปู	crab	ဖွန်း
115	อีกา	crow	ကျီးကန်း
116	สุนัข	dog	ခွေး
117	นกพิราบ	dove	ချိုးငှက်
118	เป็ด	duck	ဝမ်းဘဲ

119	นกอินทรีย์	eagle	လင်းယန်
120	ช้าง	elephant	ဆင်
121	สุนัขจิ้งจอก	fox	မြေခွေး
122	กวาง	deer	သနမင်
123	แพะ	goat	ဆိတ်
124	ห่าน	goose	ဂန်း
125	หงส์	cob	ရေဤဂန်း
126	กระต่าย	rabbit	ယုန်
127	เนื้อทราย	deer	ရယ်
128	เขา	horn	ဦးချို
129	เสือ	tiger	ကျား
130	พังพอน	mongoose	ခွေးပါ
131	ลิง	monkey	မျောက်
132	หนู	rat	ကြွက်
133	นาก	otter	ဖျံ
134	หอย	shell	ခရု
135	นกแก้ว	popinjay	ကျက်တူရွေး
136	ปลาตะเพียน	carp	ငွေ့ကင်း
137	ปลาช่อน	kind of freshwater catfish	ငါးခူ
138	ปลาฉลาม	shark	ငါးဆင်

139	ปลาหมอ	climbing perch	ငါးပြေမ
140	ปลาตาพอง	river catfish	ငါးတန်
141	แมงกะพรุน	jellyfish	ငါးထီး
142	ปลากะพง	flying fish	ငါးပျံ
143	ปลาไหล	eel	ငါးရှဉ်း
144	ปลาวาฬ	whale	ငါးဝန်
145	นกยูง	peacock	ဒေါင်း
146	นกกระยาง	egret	ဗျိုင်း
147	เหยี่ยว	hawk	ဥမ်းငှက်
148	นกฮูก	owl	ဇီကွက်
149	นกกระจอก	sparrow	စာကလေး
150	นกปากซ่อม	snipe	ခမြုံပ်ငှက်
151	นกนางแอ่น	swallow	မိုးဇွေငှက်
152	ไก่ฟ้า	pheasant	ရစ်
153	ไก่งวง	turkey	ငှက်ဆင်
154	อีแร้ง	buzzard	လင်းတ
155	วัวป่า	aggregation	စိုင်
156	เม่น	hedgehog	ပဖို
157	มด	ant	ပရွက်ဆိတ်
158	ปลวก	termite	ခြ

159	ผึ้ง	bee	ပျား
160	เรือด	bedbug	ှကမ်းပိုး
161	ผีเสื้อ	butterfly	လိပ်ပြာ
162	ด้วง , ดักแด้	weevil	ခူ
163	ตะขาบ	centipede	ကင်းခြေများ
164	จระเข้	crocodile	မိကျောင်း
165	หิ่งห้อย	firefly	ပိုးစုန်းဖူ
166	หมัด , ไร (ไก่)	flea	
167	แมลงวัน	greenbottle fly	ယင်ကောင်
168	ยุง	mosquito	ခြင်
169	กบ	frog	ဖါး
170	จิ้งจอก	fox	အိမ်ခွေမွှာင်
171	ตุ๊กแก	gecko	တောက်တဲ့
172	แมลง	bug	ပိုးကောင်
173	ปลิง	leech	ခွမှာ့
174	ทาก	slug	ကွတ်
175	แมงป่อง	scorpion	ကင်းမြီးကောက်
176	ตัวไหม	silkworm	ပိုး
177	งู	snake,	ဆွမ
178	งูพิษ	poisonous snake	ဆွမဆိုး

179	งูเห่า	cobra	ဆွမဟောက်
180	งูเหลือม	Python reticulatus	စပါးကျိုး
181	งูหลาม	python	စပါးခုံး
182	แมงมุม	spider	ပင့်ကူ
183	จักจั่น	cicada	ပုဇင်းရင်ကွဲ
184	แมลงปอ	dragonfly	ပုစဉ်းကောင်
185	แมลงสาบ	roach	ပိုးဟပ်
186	ตั๊กแตน	grasshopper	နံကောင်
187	กิ้งกือ	millepede	ပိုးနသန်းကောင်
188	เหา	louse	သန်း
189	แตน	hornet	ပိတုံး
190	หนอนกะสือ	glowworm	လောက်လမ်း
191	แมลงเม่า	mayfly	ပလူ
192	ทุเรียน	durian	ဒူရင်း
193	กล้วย	banana	ငှက်ပျောသီး
194	ตาล	Toddy palm	ထန်းသီး
195	มะม่วง	mango	သရက်သီး
196	มะพร้าว	cocoanut	အုန်းသီး
197	มังคุด	mangosteen	မင်းကွတ်

198	มะขาม	tamarind	မန်ကျည်းသီး
199	ละมุด	sapodilla	သွေးသီး
200	ลิ้นจี่	lichee	တရုတ်ဇီးသီး
201	มะปราง	Marian plum	မရမ်းသီး
202	มะเฟือง	carambola	ဆေးငုံးဝါးသီး
203	ส้มโอ	grapefruit,pomelo	ပွဲကောသီး
204	ส้มเขียวหวาน	tangerine	လိမေ္မာသီး
205	ส้มเกลี้ยง	preserved citrus fruit pulp	ရောက်သီးဆေးပြား
206	มะละกอ	papaya	သသေ်ာသီး
207	สับปะรด	pineapple	နာနပ်သီး
208	ทับทิม	ruby	သလဲသီး
209	ฝรั่ง	guava	မာလကာသီး
210	ชมพู่	rose apple	နှင်းသီး
211	องุ่น	grape	သဗျက်သီး
212	แอปเปิ้ล	apple	သစ်သော့သီး ∴ ပန်းသီး
213	แตงโม	watermelon	ဖရဲသီး
214	แพร์	pear	သစ်တော်သီး
215	อินทผาลำ	date-plam	စွန်ပလွံသီး
216	มะนาว	lemon	သံပုရာသီး
217	พุทรา	monkey apple	ဇီးသီး

218	น้อยหน่า	custard apple	ဩဇာသီး
219	มะยม	star gooseberry	ဇီးပူသီး
220	อินทผาลำ	date-plam	စွန်းပလွံသီး
221	มะกรูด	kaffir lime	မကျူပ်သီး
222	มะไฟ		စန်းစောသီး
223	ลำไย	longan	ကောင်မျက်လုံးသီး
224	ขนุน	jackfruit	ပို့ဩသီး
225	มะงั่ว	citron	ရှောက်သခွားသီး
226	มะเดื่อ	sycamore	သဖန်းသီး
227	เปลือก (ไม้)	bark	အခေါက်
228	เปลือก (ผลไม้)	hull,rind	ခွံ
229	ดอก , ดวง	burgeon	အပွင့်
230	ดอก	flower	ပန်း
231	ผล , ลูก	fruit	အသီး
232	แก่น	heartwood	အနှစ်
233	ใบ	leafage	အရွက်
234	ต้นไม้	tree	အပင်
235	ลำต้น	trunk	ပင်စည်
236	ราก	rhizome	အမြစ်
237	เมล็ด , เม็ด	seed	အစေ့

238	ก้าน	limb	အရိုး
239	ขั้ว	stem of fruit	အခြေ
240	สีดำ	black	မည်း ∴ နက်ရောင်
241	สีน้ำเงิน	blue	ပြာရောင်
242	สีเทา	grey	မွဲပြာရောင်
243	สีน้ำตาล	brown	ခြေရောင်
244	สีแดงเข้ม	lake	ရဲရဲနီရောင်
245	สีแดง	red	နီရောင်
246	สีแดงเลือดหมู	crimson	သကော်သွေးရောင်
247	สีชมพู	pink	ပန်းရောင်
248	สีเขียว	green	စိမ်းရောင်
249	สีขาว	white	ယုရောင်
250	สีเหลือง	yellow	ဝါရောင်
251	สีส้ม	orange	လိမေ့ဉီရောင်
252	สีม่วงคราม	violet	ခရမ်းရောင်အရင့်
253	สีม่วง	purple	ခရမ်း
254	เช้า	cockcrow,morn	မနက် ∴ နံနက်
255	เที่ยง	midday	နေမွန်းတည့်
256	บ่าย	afternoon	နေမွန်းလွဲ
257	กลางวัน	day	နေ၌လယ်

258	ค่ำ	dark	ညနေ
259	กลางคืน	night	ည
260	เที่ยงคืน	midnight	သန်းခေါင်
261	รุ่งอรุณ , เช้า	dawn	မိုးလင်းစ ∴ စောစော
262	วัน (๒๔ ชั่วโมง)	all day long	ရက်
263	วัน (๑๒ ชั่วโมง)	daytime	နေ့
264	วันนี้	today	ယနေ့ ∴ သည်နေ့
265	พรุ่งนี้	tomorrow	နက်ဖြန်
266	มะรืน	day after tomorrow	သဘက်
267	เมื่อวาน	yesterday	မနေ့က
268	เมื่อวานซืน	the day before yesterday	တမျှန်နေ့
269	วันอาทิตย์	Sunday	တနင်္ဂနွေနေ့
270	วันจันทร์	Monday	တနင်္လာနေ့
271	วันอังคาร	Tuesday	အင်္ဂါနေ့
272	วันพุธ	Wednesday	ဗုဒ္ဓဟူးနေ့
273	วันพฤหัสบดี	Thursday	ကြာသပတေးနေ့
274	วันศุกร์	Friday	သောကြာနေ့
275	วันเสาร์	Saturday	စနေနေ့
276	เดือน	month	လ
277	ข้างขึ้น	waxing moon	လဆန်း

278	ข้างแรม	waning moon	လဆုတ်
279	ขึ้น ๑๕ ค่ำ	the fifteenth day of awaxing moon	လပြည့်
280	แรม ๑๕ ค่ำ	the fifteenth day of a waning moon	လကွယ်
281	วันโกน	Thai definition	အဖိတ်နေ့
282	วันพระ	Buddhist holy day	ဥပုသ်နေ့
283	เดือน ๕ (ปีใหม่)	the fifth lunar month	တံခူးလ
284	เดือน ๖	the sixth lunar month	ကဆုန်လ
285	เดือน ๗	the seventh lunar month	နယုန်လ
286	เดือน ๘	the eight lunar month	ဝါဆိုလ
287	เดือน ๙	the ninth lunar month	ဝါခေါင်လ
288	เดือน ๑๐	the tenth lunar month	တော်သလင်းလ
289	เดือน ๑๑	the eleventh lunar month	သတင်းကြတ်လ
290	เดือน ๑๒	the twelfth lunar month	တံဆောင်မုန်းလ
291	เดือนอ้าย	the first lunar month	နတ်တော်လ
292	เดือนยี่	the second lunar month	ပြာသိုလ်လ
293	เดือน ๓	the third lunar month	တပ်တွဲလ
294	เดือน ๔	the fourth lunar month	တပေါင်းလ
295	มกราคม	January	ဇွန်ဝါရီလ
296	กุมภาพันธ์	February	ဖေဖော်ဝါရီလ
297	มีนาคม	March	မတ်လ

298	เมษายน	April	ဧပြီလ
299	พฤษภาคม	May	မေလ
300	มิถุนายน	June	နယုန်လ
301	กรกฎาคม	July	ဇူလိုင်လ
302	สิงหาคม	August	သဂုတ်လ
303	กันยายน	September	စက်တင်ဘာလ
304	ตุลาคม	October	အောက်တိုဘာလ
305	พฤศจิกายน	November	နိုဝင်ဘာလ
306	ธันวาคม	December	ဒီဇင်ဘာလ
307	ฤดู	[N] climate; weather;season	ဥတု ∴ ရာသီဥတု
308	ฤดูร้อน	summer	နွေဥတု
309	ฤดูฝน	rainy, season	မိုးဥတု
310	ฤดูหนาว	winter; cold season	ဆောင်းဥတု
311	เวลาพระอาทิตย์ขึ้น	sunrise	နေထွက်ကာလ
312	เวลาพระอาทิตย์ตก	sunset	နေဝင်ကာလ
313	สว่าง	light	လင်း
314	มืด	dark	ချုပ်
315	ปี	year	နှစ်
316	เรือน (ไม้) บ้าน	house	အိမ်
317	บ้าน (ตึก)	building	တိုက်

318	พื้นบ้าน	folk	ကြာမ်းခင်း
319	เสา	bollard	တိုင်
320	ฝา	wall	နံရံ
321	หลังคา	roof	အမိုး
322	ประตู	door	တံခါး
323	หน้าต่าง	window	ပုတင်းပေါက်
324	เพดาน , ฝ้า	ceiling	မျာနှာကြက်
325	อาคาร	apartment	အဆောက်အဦ
326	จังหวัด	province	ခရိုင်
327	อำเภอ	district	မြို့. နယ်
328	ตำบล	point	ရပ်ကွက်
329	พืชผล	crops	အသီးအနှံ
330	ไร่นา	a paddy field	လယ်ယာ
331	ชาวนา	farmer	လယ်သမား
332	คราด , ไถ	pinchfork	ထွန် ∴ ထယ်
333	รถไถ	pushcart	ထွန်စက်
334	แปลงนา , แปลงสวน	paddy ,fieldbotanical garden	စိုက်ပျိုးခင်း
335	ปลูก	bed out	စိုက်
336	สวน	garden	ဥယျာဉ်
337	สนามหญ้า	sward, lawn	မြက်ခင်း

338	ยุ้งข้าว	barn	စပါးကျီ
339	กระท่อมข้าวเปลือก	barn,granary	စပါးပုတ်
340	รวงข้าว	spike	စပါးနှံ
341	ลานนวดข้าว	rice straw	တလင်း
342	ฟางข้าว	straw	ကောက်ရိုး
343	ต้นข้าว	rice	ကောက်ပင် ∴ စပါးပင်
344	ปุ๋ย	nitrate of sodium	မြေဩဇာ
345	ปุ๋ยเคมี	chemical fertilizer	ဓါတ်မြေဩဇာ
346	รหัส(วิดน้ำ)	drop of water	ရေစက်
347	น้ำจืด	freshwater	ရေချို
348	น้ำเค็ม	brine	ရေငန်
349	การเก็บเกี่ยว	bring in	ကောက်ရိတ်ခြင်း
350	การเพาะปลูก	cultivate	စိုက်ပျိုးရေး
351	เคียว	sickle	တံစဉ်
352	บิดา	father	ဖခင် ၊ အဖေ
353	มารดา	mater	မိခင် ၊ အမေ
354	ครู , อาจารย์	teacher	ဆရာ
355	ญาติพี่น้อง	kinsfolk	အသိမိတ်ဆွေနှင့်ဆွေမျိုးစပ်
356	ปู่ , ตา	granddad	အဘိုး
357	ย่า, ยาย	grandmama, grandmother	အဘွား

358	ลุง	uncle	ဘွကီး
359	ป้า	aunt	ကီးတော်
360	น้า (ชาย , หญิง)	aunt	ဦးလေး ။ အကျိုး
361	อา (ชาย , หญิง)	mother younger brother	ဦးလေး ။ အကျိုး
362	พี่ชาย	brother	အစ်ကို ∴ အကို
363	น้องชาย	younger brother	ညီလေး
364	พี่สาว	elder sister	အမ
365	น้องสาว	sister	အစ်မ ∴ ညီမ
366	เด็กทารก	neonate	နို့စို့ကလေး
367	บุตรชาย	son	သား
368	บุตรสาว	daughter	သမီး
369	ลูกเขย	son-in-law	သားမက်
370	ลูกสะใภ้	daughter-in-law	ချွေးမ
371	พ่อตา	the father of one's wife	ယောကျ်ာမ
372	แม่ยาย	mother-in-law	ယောကျ်ာမမိန်းမ
373	เด็กชาย	boy	ယောကျ်ားကလေး
374	เด็กหญิง	girl	မိန်းကလေး
375	หลานชาย , หลานสาว	grandson,granddaughte	တူ ∴ တူမ
376	สามี	husband	လင်
377	ภรรยา	wife	မယား

378	ชายหนุ่ม (โสด)	bachelor	လူပျို
379	หญิงสาว (โสด)	maid	အပျို
380	การแต่งงาน	marriage	လက်ထက်ခြင်း
381	ชายที่แต่งงานแล้ว	the head of the family	အိမ်ထောင်ယောက်ျား
382	หญิงที่แต่งงานแล้ว	housewife	အိမ်ထောင်မိန်းမ
383	พ่อเฒ่า	grandsire	အဖို့ကီး
384	แม่เฒ่า	gammer	အမွေကီး
385	บิดามารดา	legitimate	မိဘ
386	บิดาเลี้ยง	stepfather	ဘထွေး
387	มารดาเลี้ยง	stepmother	မိထွေး
388	บุตรเลี้ยง	stepchild	အထက်သား
389	พ่อหม้าย	widower	မုဆိုးဖို
390	แม่หม้าย	widow	မုဆိုးမ
391	ผม (เส้นผม)	hair	ဆံပင်
392	ศีรษะ	head	ဦးခေါင်း
393	กะโหลกศีรษะ	skull	ဦးခွံ
394	หน้าผาก	forehead	နဖူး
395	หู	ear	နား
396	ใบหู	earlobe	နားရွက်
397	คิ้ว	eyebrow	မျက်ခုံး

398	ตา	eye	မျက်စိ
399	จมูก	nose	နှာခေါင်း
400	ปาก	mouth	နှုတ်∴ ပါးစပ်∴ ခံတွင်း
401	แก้ม	cheek	ပါး
402	คาง	chin	မေးစေ့∴ စည့်
403	คอ	neck	လည်ပင်း
404	ไหล่	shoulder	ပုခုံး
405	มือ	hand	လက်
406	แขน	arm	လက်မောင်း
407	ข้อศอก	elbow	တံတောင်ဆစ်
408	หน้าอก	breast	ရင်ပတ်
409	ท้อง	abdomen	ဝမ်းပိုက်
410	เอว	waist	ခါး
411	สะโพก	haunch	တင်ပါး
412	ขา	leg	ခြေထောက်
413	เท้า	foot	ခြေ
414	ฝ่าเท้า	sole	ခြေဖဝါး
415	หลังเท้า	instep	ခြေဖမိုး
416	นิ้วเท้า	toe	ခြေချောင်း
417	นิ้วมือ	finger	လက်ချောင်း

418	ฝ่ามือ	palm	လက်ဖဝါး
419	หัวเข่า	knee	ဒူး
420	สีข้าง	flank	နံဘေး
421	ลิ้น	tongue	လျှာ
422	ฟัน	tooth	သွား
423	กราม , ฟันกราม	molars	အန် ∴ အန်သွား
424	เขี้ยว	canine tooth	စွယ် ∴ စွယ်သွား
425	ฟันหน้า	front teeth	ရှေ့ခွံသွား
426	ลูกกะเดือก	Adam's apple	လည်စေ့ ∴ စည့်
427	มันสมอง	brain	ဦးနှောက်
428	เลือด	blood	သွေး
429	กระดูก	bone	အရိုး
430	เส้นประสาท	nerve	အာရုံကြော
431	อวัยวะเพศหญิง	gonad	ယောကျ်ားအင်္ဂါဇာတ်
432	อวัยวะเพศชาย	penis	မိန်းမအင်္ဂါဇာတ်
433	ร่างกาย	body	ကိုယ်ကာယ ∴ ခန္ဓာ ∴ ကိုယ်
434	หัวใจ	heart	နှလုံး
435	ไต	wen	ကျောက်ကပ်
436	ปอด	lung	အဆုတ်
437	ตับ	liver	အသည်း

438	กระเพาะอาหาร	stomach	အစာအိမ်
439	ไส้	intestine	အူ
440	อุจจาระ	crap	ကျင်ကြီး ၊ မစင်
441	ปัสสาวะ	spend a penny	ကျင်ငယ် ၊ ပေါ့
442	ขน	goose	အမွေး
443	กล้ามเนื้อ	brawn,biceps	့ကက်သား
444	นิ้วชี้	index finger	လက်ညိုး
445	นิ้วนาง	ring finger	လက်သူကြိုယ်
446	นิ้วกลาง	middle finger	လက်ခလယ်
447	นิ้วก้อย	pinkie	လက်သန်း
448	อายุ	age	အသက်အရွယ်
449	อายุมาก	up in years,old	အသက်ကီး
450	อายุยืน	long life	အသက်ရှည်
451	ความโกรธ	anger	အမျက် ၊ ဒေါသ
452	วิทยาการ	technology	အတတ်ပညာ
453	ความกว้าง	width	အနှ
454	บุคลิก	identity	အသရေ
455	ความเป็นเด็ก	childhood	သူငယ်အဖြစ် ၊ဘာဝ
456	ความลึก	deepness	နက်ခြင်း
457	ความไม่ชอบใจ	content	မနှစ်သက်ခြင်း

458	จิต , ใจ	mind	သဘော ၊ စိတ်
459	ความกังวลใจ	worry	စိုးရိမ်ခြင်း
460	ความหลง , ความโง่	crassitude	မောဟ ၊ မိုက်ခြင်း
461	ความนุ่มนวล	gentleness	သိမ်မွေ့ခြင်း
462	ความดี	niceness	ကောင်းမှု
463	ความใหญ่	big	ကြီးခြင်း
464	ความเกลียดชัง	animosity	မုန်းတီးခြင်း
465	ความสูง	height	အရပ်မြင့်ခြင်း
466	ความซื่อตรง	naivety	စိတ်မြင့်ခြင်း
467	ความมีเกียรติ	dignity	ဂိုဏ်အသရေရှိခြင်း
468	ความดีใจ	gladness	ဝမ်းမြောက်ခြင်း
469	ความสามารถในการดำเนินการ	capability	ဆင်ခြင်ရန်သတိမူ
470	ความรู้	acquaintance	အသိပညာ
471	การหัวเราะ	laughing	ရယ်ခြင်း
472	ความยาว	length	အရှေ ၊ အလျား
473	ความรัก	love	ချစ်ခြင်း
474	ความอดทน , ขันติ	enduringness	သည်းခံခြင်း
475	ความสุภาพ	civility,comity	ယဉ်ကျေးခြင်း
476	เหตุผล	reason	အကြောင်းယုတိမူ
477	ความระลึกชาติ	recall one's former incarnation	ဝေဒနာ

478	ความรู้สึก	aesthesia	ခွေ့တွထိခြင်း
479	การได้ยิน	hearing	နာ့ကားခြင်း
480	การได้เห็น	sight for sore eyes	မြင်ခြင်း
481	การได้กลิ่น	scent	အနံ့ခံရခြင်း
482	รสชาด	savour	အရသာ
483	รูปร่าง	figure	ပုံသ့ဉ်ာန်
484	ขนาด	size	အရွယ်
485	กลิ่น	flavour	အနံ့
486	การยิ้ม	smiling	ပုံးခြင်း
487	ความเสียใจ	ruth	ဝမ်းနည်းခြင်း
488	พลังกาย	sinew	ခွန်အား
489	ความเซ่อ , ความไม่ฉลาด	softheaded	ဥဉ်ထုံးခြင်း
490	รู้รส	facial nerve	အ့ံသခြင်း
491	ความหนา	layer	ထူခြင်း
492	ความคิด	spectrum	စဉ်းစားခြင်း
493	เสียง	noise	အသံ
494	ความอ่อนแอ	anemia	အားနည်းခြင်း
495	ความฉลาด	acuity	လိမ္မာခြင်း
496	ความแข็งแรง	stamina	အာ့ကီးခြင်း
497	โรงพยาบาล	hospital	ဆေးရုံ

498	หมอ (นายแพทย์)	doctor	ဆရာဝန်
499	นางพยาบาล	nurse instructor	သူနာပုဆရာမ
500	พยาบาล	nurse	သူနာပု
501	รถพยาบาล	ambulance	သူနာတင်ကား
502	ผ้าพันแผล	bandage	အစည်းအပတ် ∴ ပတ်တီး
503	เข็มฉีดยา	hypodermic syringe	ဆေးထိုးအပ်
504	โรค	affection	ရောဂါ ∴ အနာ
505	ห้องตรวจโรค	diagnoserome	ရောရါစပ်းစပ်ခန်း
506	ตรวจ	diagnose	စမ်းစပ်
507	บัตรคนไข้	patient's record	လူနာလက်မှတ်
508	ห้องจ่ายยา		ဆေးပေးခန်း
509	ใบสั่งยา	recipe	ဆေးညွှန်းစာ
510	ผ่าตัด	operate	ခွဲစိတ်
511	ห้องผ่าตัด	operating room	ခွဲစိတ်ခန်း
512	นายแพทย์ผ่าตัด	surgeon	ခွဲစိတ်ဆရာဝန်
513	คลอด	bear,deliver of	သားဖွါး ∴ ကလေးမွေး
514	ห้องคลอด	delivery room	သားဖွါးခန်း
515	นางผดุงครรภ์	midwife	သားဖွါးဆရာမ
516	สละเลือด	donor	သေးလ၍
517	โดนงูกัด	snakebite	ဓ္ဍမကိုက်ခံရသည်

518	เซรุ่ม	serum	ဆွမဆိပ်အရည်
519	ฉีด (ยา)	inject	ထိုးသည်
520	ปลูกฝี	inoculate against	ကျောက်ရောဂါပျိုးပေးသည်
521	ยา	medico	ဆေး ၊ ဆေးဝါး
522	ยาเม็ด	pill	ဆေးလုံး
523	ยาน้ำ	potion	ဆေးရည်
524	ยากิน	oral drug	စားဆေး
525	ยาทา	liniment	လိမ်းဆေး
526	ยาฉีด	injection	ထိုးဆေး
527	แค้ปซูล	capsule	ကက်စွန်လ်
528	เอ็กซ์เรย์	encephalogram	ဓါတ်မှန်
529	ฉายเอ็กซ์เรย์	treat by radiation	ဓါတ်မှန်ရိုက်
530	ปวดศรีษะ	have a headache	ခေါင်းကိုက်
531	ปวดท้อง	stomach ache	သွားကိုက်
532	ปวดข้อ	dengue	အဆစ်အမြစ်ကိုက်ခဲ
533	ปวดหลัง	have a backache	နောက်ကျောတက်
534	ปวดเอว	lumbago	ခါးနာ
535	ตัวร้อน	run a temperature	ကိုယ်ပူ
536	เป็นไข้	feverish	အဖျားတက်
537	หนองฝี	pock	ပြည်ရည်

538	เป็นโรคฝีดาษ	smallpox	ကျောက်ရောဂါဖြစ်သည်
539	โรคผิวหนัง	impetigo	အသားအရေရောဂါ
540	ขี้กราก	ringworm	ပွေ့ ရောဂါ
541	ขี้เกลื้อน	chloasma	တုတ်ရောဂါ
542	หิด	scabies	ယားရောဂါ
543	อหิวาตกโรค	cholera	ကာလဝမ့်ရောဂါ
544	โรคปอด	pneumonia	အဆုတ်နာရောဂါ
545	โรคตับ	liver disease	အသည်းရောင်ရောဂါ
546	โรคมะเร็ง	cancer	ကင်ဆာရောဂါ
547	โรคเรื้อน	leprosy	အနာကြီးရောဂါ
548	โรคกระเพาะอาหาร	gastritis	အစာအိမ်ရောဂါ
549	กามโรค	venereal disease	ကာလသား ရောဂါ
550	โรคผู้หญิง	venereal disease	အပျိုနာရောဂါ
551	โรคหัวใจ	heart disease	နှလုံးရောဂါ
552	อีสุกอีใส	varicella	ကျောက်ဖွဖုနာ
553	หวัด	cold	နှာစည်း
554	ท้องร่วง	have diarrhea	ဝမ်းကျ
555	อ่อนเพลีย	wayworn	အားကုန်ခန်းခြင်း
556	หัด	measles	ဝက်စသက်နာ
557	เป็นลม	faint,black out	လေတက်နာ ∴ တက်နာ

558	อัมพาต	paralysis	လေသင်တုန်း
559	ริดสีดวง	hemorrhoid	မြင်းသရိုက်နာ ∴ လိပ်ခေါင်းနာ
560	ยาบำรุง	tonic	အားတိုးဆေး
561	อาเจียน , คลื่นไส้	bring up,fetch up	အန်
562	หาย(จากการเจ็บป่วย)	be on one's feet	ပျောက်ကင်း
563	ทุเลา	recover	သက်သာ
564	ตาย	die	သေ ∴ ဆုံး ∴ ကွယ်လွန်
565	อาหาร	food	အစာအဟာရ
566	ของกิน	foodstuff	စားစရာ
567	ข้าวสวย	rice	ထမင်း
568	ข้าวเหนียวนึ่ง	sticky rice	ကောက်ညှင်း
569	ข้าวหลาม	glutinous rice roasted in bamboo joints	ကောက်ညှင်းကျည်တောက်
570	ข้าวต้มมัด	stemed glutinouts rice	ကောက်ညှင်းပေါင်း
571	ข้าวเหนียวดำ	black sticky rice	ကောက်ညှင်း
572	ข้าวเหนียวดำนึ่ง	purplish black variety of glutinous rice	ကောက်ညှင်းချိုပ်
573	ข้าวตอกปั้นกลมผสมมะพร้าวและน้ำตา	pop-corn rilled into a ball with palm sugar	ပေါက်ပေါက်ဆုပ်
574	ข้าวเหนียว	glutinous rice	ကောက်ညှင်းဆန်
575	ข้าวจ้าว	round shaped rice	ဆန်
576	ข้าวผัด	fried rice	ထမင်းကြော်
577	ข้าวต้ม	rice gruel	ဆန်ပြုတ်

578	แกง	curry	ဟင်း
579	ผัด , ทอด	fry	ကျော်
580	แกงเนื้อ	julienne	အမဲသားဟင်း
581	แกงหมู	green curry	ဝက်သားဟင်း
582	แกงไก่	mussaman curry	ကြက်သားဟင်း
583	ขนมจีน	rice noodles	မုန့်တီ
584	ขนมจีนพม่า	rice noodles with gravy	မုန့်တီဟင်းခါး
585	ย่าง	bake	ကင်
586	เนื้อย่าง	barbecue	အမဲသားကင်
587	ไก่ย่าง	roasting chicken	ကြက်သားကင်
588	กุ้งแห้ง	dried shrimp	ပုစွန်ခြောက်
589	น้ำปลาหวาน		ချိုတဲ့ငံပြာရည်
590	สะเดา	nim	တမာ
591	น้ำซุบ	broth	အူပုတ်ရည်
592	ก๋วยเตี๋ยวน้ำ	noodle	ခေါက်ဆွဲပုတ်
593	ก๋วยเตี๋ยวแห้ง	noodle	ခေါက်ဆွဲဆီခွက်
594	สลัด	green salad	ဟင်းရွက်လက်သုပ်
595	เบคอน	gammon	ဝက်သားကပ်တိုက်
596	ไส้กรอก	bologna	ဝက်အူချောင်း
597	ยำ	mix	လက်သုပ်

598	ยำวุ้นเส้น	thick rice noodle	ကြာဇံလက်သုပ်
599	ยำเนื้อ	meat salad	အမဲဘားလက်သုပ်
600	เครื่องดื่ม	beverage	သောက်စရာ
601	น้ำอัดลม	soft drink	ဘိလပ်ရေ
602	น้ำกลั่น	distilled water	အစက်ချရေ
603	นม , นมวัว	milk	နွားနို့
604	สุรา	alcohol	အရက်
605	เบียร์	beer	ပီယာ ။မြို့
606	วิสกี้	whisky	ဝစ်စကီအရက်
607	กะแช่	(n) palm wine	အုန်းထန်းခနိပေရေ
608	ของหวาน	dessert	အချိုအချဉ်
609	ขนม	food other than main meal	မုန့်
610	น้ำแข็ง	ice	ရေခဲ
611	โซดา	soda	ဆိုဒါ
612	ขนมเค็ก	cake	ကိတ်မုန့်
613	ขนมปังแผ่น	bread	ပေါင်မန့်
614	น้ำผึ้ง	honey	ပျားရည်
615	น้ำตาลปอนด์	cube sugar	သကြားခဲ
616	น้ำตาลทราย	sugar	သကြား
617	แยม	jam	သစ်သီးယို

618	กวน	churn to	ယို
619	ทุเรียนกวน	to preserve durian by boiling	ုးရင်းသီယို
620	มะม่วงกวน	to preserve mango by boiling	သရက်သီယို
621	น้ำเชื่อม	sirup	သကြားရည်အဖြစ်
622	เครื่องดูด , สูบ	lift pump	သောက်စရာ
623	บุหรี่	cigaret	ဆေးလိပ်
624	ดูด , สูบ (บุหรี่)	chain-smoke	သောက်
625	ดูด , สูบ (ฝิ่น , กัญชา)	chong it	ရှူ
626	กัญชา	cannabis	ဆေးခြောက်
627	ฝิ่น	opium	ဘိန်း
628	ซิการ์	cigar	ဆေးပြင်းလိပ်
629	ซิกาเร็ท	cigarette	စီကာရက်
630	บุหรี่ก้นกรอง	filter	အဆိခံဆေးလိပ်
631	ไปค์ , กล้อง	pipe,peace pipe	ဆေးတံ
632	น้ำปลา	fish sauce	ငန်ပျာရေ
633	เกลือ	salt	ဆား
634	ปลาร้า	pickled fish	ငါးပိရေ
635	ปลาแห้ง	dried snakeheads	ငါးခြောက်
636	ของดอง	preserve	သနပ်
637	ซอส	sauce	စမဲ

638	สุก	cooked	ကျက်
639	สุก ๆ ดิบ ๆ	half-cooked	မကျက်တကျက်
640	อาหารเช้า	breakfast	မနက်စာ
641	อาหารกลางวัน	lunch	နေ့လည်စာ
642	อาหารเย็น	dinner	ညနေစာ
643	เงิน	money	ကျပ်
644	สตางค์	satang	ပြား
645	หนึ่งสลึง	a quarter of a baht; a twenty -five satang piec	တမတ်
646	สองสลึง	a half of a baht; a fifty satang piece	ငါးမူး
647	สามสลึง	a seventy -five satang piece	သုံးမတ်
648	หนึ่งจั๊ต		တကျပ်
649	หกสลึง	six satang piece	တကျပ်ခွဲ
650	เจ็ดสลึง	seven satang piece	တကျပ်ခုနစ်ဆယ်ငါးပြား
651	สองจั๊ต		နှစ်ကျပ်
652	สามจั๊ต		သုံးကျပ်
653	ยี่สิบห้า	a twenty -five	အစီတ်
654	๑	one	တစ် ၊ ၁
655	๒	two	နှစ် ၊ ၂
656	๓	three	သုံး ၊ ၃
657	๔	four	လေး ၊ ၄

658	๕	five	ငါး ၊ ၅
659	๖	six	ခြောက် ၊ ၆
660	๗	seven	ခုနစ် ၊ ၇
661	๘	eight	ရှစ် ၊ ၈
662	๙	nine	ကိုး ၊ ၉
663	๑๐	ten	တဆယ် ။ ဆယ် ၊ ၁၀
664	๑๑	eleven	ဆယ်တစ် ၊ ၁၁
665	๑๒	twelve	ဆယ်နှစ် ၊ ၁၂
666	๒๐	twenty	နှစ်ဆယ် ၊ ၂၀
667	๒๑	twenty one	နှစ်ဆယ်တစ် ၊ ၂၁
668	๓๐	thirty	သုံးဆယ် ၊ ၃၀
669	๓๒	thirty two	သုံးဆယ်နှစ် ၊ ၃၂
670	๔๐	forty	လေးဆယ် ၊ ၄၀
671	๕๐	fifty	ငါးဆယ် ၊ ၅၀
672	๖๐	sixty	ခြောက်ဆယ် ၊ ၆၀
673	๗๐	seventy	ခုနစ်ဆယ် ၊ ၇၀
674	กั้น	block	ကာ
675	รถ	car	ကား
676	รักษา	give medical treatment	ကု
677	ช่วย	help	ကူညီ

678	คัดลอก	copy	ကူး
679	ข้ามน้ำ	cross over	းကူ
680	คำอุทาน (เอาละ)	interjection	
681	ยึด	silt up	ကော
682	กาว , งัด	glue	ကော်
683	กรรม	one s deed	ကံ
684	โชคดี	be lucky	ကံကောင်းတယ်
685	กราบไหว้	worship	ကန်တော့
686	ดอกบุญนาค		းကံကောင်ပန်
687	ท้องฟ้า	canopy	ကောင်းကင်
688	ไก่ตัวผู้	cock	ကြက်ဖ
689	นับถือบูชา	adore	ကိုးကွယ်
690	จับ , ถือ	carry	ကိုင်
691	มั่นคง , คงทน	constant	ကြံ့ခိုင်
692	โค้ง	arched	းကုန်
693	เติบโต	grow up	းကြီးထွာ
694	ท้องทุ่ง	grassland	ကွင်းပြင်
695	ประโยชน์	use	းကိုယ်ကျို
696	เกาะ	ait	းကျွန်
697	สวนครัว	kitchen garden	းကိုင်

698	ตก , ถึง (เวลา)	drop out	ကျ
699	เสื้อ	blouse	းကျာ
700	รำ	dance	က
701	กาแฟ	heart disease	ကော်ဖီ
702	กลม	varicella	ကော်ဇော
703	โชคดี	cold	းကံကောင်
704	ทองแดง	have diarrhea	ကျးနီ
705	หิน	wayworn	ကျောက်
706	เหล็ก , สมอเรือ	measles	းကျောက်ဆူ
707	ไก่แจ้	faint,black out	ကြာက်တည်ငိ
708	ได้ยิน	paralysis	ရကြံေ
709	นาน	hemorrhoid	းကြာ
710	บัวตูม	tonic	းကြာဖူ
711	คอกบัว	bring up,fetch up	ကြာ
712	น่ากลัว	be on one's feet	ကြောက်စရာ
713	แมวน้อย	recover	းကြောင်ကလေ
714	ชอบ	die	ကြိုက်
715	ต้อนรับ	food	ကြိုဆို
716	ธุระ	foodstuff	းကိစ်စမျာ
717	ท่าข้าม	rice	ကူးတို့

718	สินค้าทั้งหลาย	sticky rice	းကုန်မျာ
719	โรงเก็บสินค้า	glutinous rice roasted in bamboo joints	ကုန်လျောက်ရိ
720	พ่อค้าใหญ่	stemed glutinouts rice	းကုန်သည်ကြီ
721	เก้าอี้	black sticky rice	ကုလားထိုင်
722	ครั่นเนื้อครั่นตัว	purplish black variety of glutinous rice	ကိုက်ခဲ
723	ปลูกข้าว	pop-corn rilled into a ball with palm sugar	ကောက်စိုင်
724	กุศล	glutinous rice	ကုသိုလ်
725	ฟางข้าว	round shaped rice	းကောက်ရိ
726	ถ่านหิน	fried rice	းကျောက်မီးသွေ
727	หมาก	rice gruel	းကွမ်းသီ
728	หมากแห้ง	curry	ကွမ်းသီးခြောက်
729	ใบพลู	fry	ကွမ်းရွက်
730	ต้นสัก	julienne	ကျွန်းပင်
731	แพไม้สัก	green curry	ကျွန်းဖေါင်
732	หมูชุม	mussaman curry	ကြာက်ပေါ
733	พอใจ	rice noodles	ကျေနပ်
734	แตกต่าง	rice noodles with gravy	းကွာမြာ
735	เตรียมการ	bake	ကြိုတင်
736	จม	barbecue	ကျုံ
737	ผู้หญิง	roasting chicken	ကျားမ

738	เป็นเมือก	dried shrimp	ကျဲ
739	หัวเราะ		ကျဲကျဲ
740	ยุ้ง	nim	ကြီ
741	อีกา	broth	းကြီးကန်
742	ต้นกก	noodle	ကျ။
743	ล่วงล้ำ	noodle	းကျ။
744	ย่อย	green salad	ကျေ
745	นกแก้ว	gammon	းကျေ
746	กระจายออกไป	bologna	ကျဲ
747	จัด (แดด)	mix	ကျဲကျဲ
748	ทอง	thick rice noodle	ကျက်
749	บ่วง	meat salad	ကျော့
750	สุขภาพ	beverage	းရကျန်းမာေ
751	หลัง (อวัยวะ)	soft drink	ကျော
752	หนี้สิน	distilled water	းကြော
753	ฉาบ , แต้ม	milk	ကျဲ
754	โอ้อวด	alcohol	ကြွားဝါ
755	พานแก้ว	beer	ကြာကလပ်
756	นกแก้ว	whisky	းကြာက်တူရွေ
757	ขั่ว	(n) palm wine	းကျီစာ

758	ถ้วยทอง	dessert	ကြေးဖလာ
759	คำขวัญ	food other than main meal	ကြွာကြွေးကြော်
760	แข็งแรงอดทน	ice	ကျန်းမာကြံ့ခိုင်
761	หวาย	soda	ကြိမ်
762	ตระกร้าหวาย	cake	းကြမ်မြင်
763	ตัวเอง	bread	ကိုယ်ကိုယ်
764	สมบูรณ์	honey	းကောင်းစာ
765	พยายาม	cube sugar	းကြိုးစာ
766	ประพฤติ	sugar	းကျင်
767	เชือก 1 เส้น	jam	းကြိုးတချောင်
768	กรวย	churn to	ကတော
769	อวด	to preserve durian by boiling	းကြွာ
770	ร่วง	to preserve mango by boiling	ကွေးသည်
771	ช่างซ่อมรถ	sirup	ကားဆရာ
772	ควาย	lift pump	ကွဲ
773	ภูมิใจ	cigaret	ကြွာ
774	คาว	chain-smoke	ကြာယ်
775	เก็บ	chong it	ကောက်
776	มอง , ดู	cannabis	ကြာည့်
777	เล่น	opium	းကစာ

778	ของเล่น	cigar	ကစားစရာ
779	โรงเรียน	cigarette	းကျောင်
780	ดี	filter	းကောင်
781	หมาก	pipe,peace pipe	ကွမ်းအစ်
782	เกาะ (ทะเล)	fish sauce	းကျွန်
783	แน่น	salt	ကျစ်
784	กาชาด	pickled fish	ကြာက်ခြေနီ
785	ติด	dried snakeheads	ကပ်
786	การควบคุม	preserve	ကွပ်ကွဲမှု
787	ดู	sauce	းကျုံ
788	แข็งกระด้าง	cooked	ကြာပ်ဆတ်
789	สีมือ (ข้าว)	half-cooked	ကြိတ်
790	เกี่ยวข้าว	breakfast	ကောက်ရိတ်
791	คึกคัก	lunch	းကြာတ်ကြိတ်တို
792	บุญคุณ	dinner	းကျေးဇူ
793	ชอบ	money	ကြိုက်
794	คนโท	satang	းတကောင်
795	หยิบ , จับ	a quarter of a baht; a twenty -five satang piec	ကိုင်
796	เกา	a half of a baht; a fifty satang piece	ကုတ်
797	ร่างกาย	a seventy -five satang piece	ကိုယ်လက်

798	ออกกำลังกาย		ကျန်းမာရေးလေကျင်
799	ติด (โรค)	six satang piece	ကပ်ရောက်
800	รวย	seven satang piece	ကြွယ်
801	ริมฝั่ง		ကမ်းနားမှာ
802	สินค้า		ကုန်
803	ป้องกัน	a twenty -five	ကာကွယ်
804	ประโยชน์	one	းကောင်းကျိုး
805	ชำนาญ	two	ကျွမ်းကျင်
806	หมู่บ้าน	three	ကျေးရွာ
807	เติบโต	four	းကြီးပြင်
808	คนขับรถ	five	းကားမောင်းသမာ
809	บ้วน	six	းကျင်
810	การรักษา	seven	းရကုသ
811	พบปะ	eight	ကျင်းပ
812	ตัว , ลำตัว	nine	ကိုယ်
813	ติด	ten	ကပ်
814	(สุก ข้าว	eleven	ကျက်
815	กว้างขวาง	twelve	းကျယ်ဝန်
816	หอม	twenty	ကြက်သွန်နီ
817	กระเทียม	twenty one	ကြကသွန်ပြူ

818	จำวัด	thirty	ကွန်တော်မူ
819	ไก่ย่าง	thirty two	ကျက်ကင်
820	ครั้ง	forty	ကြိမ်
821	ไม้ตีพริก	fifty	ကျည်ပွေ
822	คัมภีร์	sixty	းကျမ်
823	ดินสอ	seventy	ခဲတံ
824	พอใจ	block	ကျေနပ်
825	ส้มป่อย	car	ကင်မွန်းချဉ်
826	เช่น	give medical treatment	ကဲ့သို့
827	เมื่อย	help	ကိုက်ခဲ
828	ฝีดาษ	copy	ကျောက်ပေါက်မာ
829	พืชผล	cross over	ကောက်ပဲသီးနှံ
830	ที่ราบสูง	interjection	ကုန်းပြင်မြင့်ဒေသ
831	ตู้กับข้าว	silt up	ဧကာင်အိမ်
832	ลูกบาค์ก	glue	ကုဗ
833	เลยไปหมด	one s deed	းကုန်ဆုံ
834	แทน	be lucky	ကိုယ်စားပြု
835	คด , งอ	worship	ကောက်
836	ตรง		ကိုက်
837	ข่าวลือ	canopy	းကောလာဟလသတင်

838	คิดในใจ	cock	းကြံစည်စတ်ကူ
839	ร่วง	adore	ကြာ
840	ทะเยอทะยาน	carry	ကြာတ်
841	ป้องกัน	constant	ကာကွယ်
842	ประกาศ	arched	ကျည၁
843	กัดขาด	grow up	ကိုက်ဖြတ်
844	ถึง	grassland	ကျရောက်
845	แซ่หวาย	use	းကိမ်လုံ
846	ตลาดโลก	ait	ကမာဈေးကွက်
847	กล้องถ่ายรูป	kitchen garden	ကယ်မ့ဒါ
848	ซองกล้องถ่ายรูป	drop out	ကယ်မ့ဒါအိတ်
849	หนังสือเรียน	blouse	ကျောင်းသုံးစာအုပ်
850	เพื่อนเล่น	dance	ကြကစား၀ေ
851	ห่าง	heart disease	ကာ
852	ต้อนรับ	varicella	ကြိုဆို
853	สัญญา	cold	းကတိ၀ေ
854	อัญมณี	have diarrhea	ကျောက်မျက်ရတနာ
855	สงสาร	wayworn	ကြာင်နာ
856	จับ	measles	ကိုင်တွယ်သုံးစွဲ
857	กระดานหิน	faint,black out	းကျောက်သင်ပုန်း

858	กระดานคำ	paralysis	းကျောင်းသင်ပုန်းကီ
859	เอาใจใส่	hemorrhoid	ပြုရဂ
860	ส่วนตัว	tonic	ကိုယ်ပိုင်
861	กระจกเงา	bring up,fetch up	ကြာည့်မှန်
862	อูฐ	be on one's feet	ကုလားအုပ်
863	ทหารรักษาพระองค์	recover	ကိုယ်ရံတော်တပ်
864	ปราศจาก	die	ကင်းအောင်
865	แบนราบ	food	ကျယ်ပြန်
866	ซุงไม้สัก	foodstuff	ကွန်းတိုင်
867	โรงสีข้าว	rice	ကြိတ်ဆုံ
868	สี	sticky rice	ကြိတ်
869	ใส	glutinous rice roasted in bamboo joints	ကြာည့်
870	ฝีดาษ	stemed glutinouts rice	ကျောက်
871	ฉีดวัคซีน	black sticky rice	းကျောက်ဆေ
872	หน้าผา	purplish black variety of glutinous rice	ကမ်းချောက်
873	ข้าวเหนียว	pop-corn rilled into a ball with palm sugar	ကောက်ညှင်းဆန်
874	กาชาด	glutinous rice	ကြာက်ခြေနီ
875	โคลง	round shaped rice	ကဗျာ
876	เจริญรุ่งเรือง	fried rice	းကြီးပွာ
877	รถบรรทุก	rice gruel	းကုန်တင်ကာ

878	ชั้นบน	curry	အပေါ်ထပ်
879	ศิลปะพื้นบ้าน	fry	ကျေးလက်အနုပညာ
880	ตกต่ำ	julienne	းကျဆင်
881	นกกระทา	green curry	ခါ
882	ขม	mussaman curry	းခါ
883	กระดึง	rice noodles	ခလောက်
884	อัน	rice noodles with gravy	ခု
885	ปุ้ง	bake	ခူ
886	เก็บ , เด็ด	barbecue	းခူ
887	อดีตกาล	roasting chicken	ခဲ့
888	แข็ง	dried shrimp	ခဲ
889	เรียก , เชิญ		ေခ
890	รับ , รอง	nim	ခံ
891	บวบ	broth	းခဝဲသီ
892	ตะบองเพชร	noodle	းခရာဆူ
893	เขา (สัตว์)	noodle	ခို့
894	ลำธาร	green salad	းချောင်
895	มะเขือ	gammon	းခရမ်းသီ
896	รอยเท้า	bologna	ခြေရာ
897	พลัง , กำลัง	mix	းခွန်အာ

898	ค่า (เงิน)	thick rice noodle	ခ
899	กีบเท้า (สัตว์)	meat salad	ခွါ
900	เดินไป	beverage	းခြေလျင်သွာ
901	ดอกทองกวาว	soft drink	းခွါညိပန်
902	ง่าม , แฉก	distilled water	ခွ
903	แปรงผม	milk	းခေါင်းဖြီ
904	เจ็บเอว	alcohol	ခါးနာ
905	ซอยผม	beer	းခွဝဲသီးခူ
906	วาน	whisky	းခိုင်
907	น่ารัก	(n) palm wine	ချစ်စရာ
908	ลูกสะใภ้	dessert	ချေးမ
909	เหงื่อออก	food other than main meal	ချွေမ
910	ไอ	ice	ချွေထွက်
911	มุ้ง	soda	ခြင်ထောင်
912	ยุงชุม	cake	ခြင်ပေါ
913	พับไว้	bread	းခေါက်ထာ
914	แขวนไว้	honey	းချိတ်ထာ
915	อวัยวะร่างกาย	cube sugar	ခြေလက်အင်္ဂါ
916	นัดหมาย	sugar	ချိန်းဆို
917	น่าชมเชย	jam	ချိန်ကျူးစရာ

918	ตาชั่ง	churn to	ချိန်ခင်
919	ผู้นำ , หัวหน้า	to preserve durian by boiling	ခေါင်းဆောင်
920	ยุง	to preserve mango by boiling	ခြင်
921	แน่น , มั่นคง	sirup	ခိုင်
922	ตระกร้า	lift pump	းခြင်
923	ปวดหัว	cigaret	ခေါင်းခုံ
924	ป้อน	chain-smoke	းခွံ့ကျွေ
925	วิ่ง	chong it	းပြေ
926	กระโดด	cannabis	ခုန်
927	เกลี้ยงเกลา	opium	ချောက်မွတ်
928	วาง	cigar	ချ
929	วางไว้	cigarette	းချထာ
930	ม้าหมุน	filter	းချာ
931	คาบ	pipe,peace pipe	ချီ
932	ชมเชย	fish sauce	းချီးကျူး
933	จาม	salt	ချေ
934	คัดค้าน	pickled fish	ချေပ
935	เช่า	dried snakeheads	းချေးငှါ
936	ขยาย	preserve	ချဲ့
937	ลื่นล้ม	sauce	ချော်လဲ

938	ประหยัด	cooked	ချွတာ
939	บกพร่อง	half-cooked	းချွတ်ယွင်
940	ด้าย	breakfast	ချည်
941	เข็มด้าย	lunch	ချည်ခင်
942	แห้ง	dinner	ခြောက်
943	ระฆัง	money	းခေါင်းလောင်
944	ตาชั่ง	satang	ချိန်ခွင်
945	แตร	a quarter of a baht; a twenty -five satang piec	ခရာ
946	ท่าน , คุณ	a half of a baht; a fifty satang piece	းခင်ဗျာ
947	กระดึง	a seventy -five satang piece	ခလောက်
948	เตะ		ခတ်
949	เลื่อย	six satang piece	ခွဲဖြတ်
950	เย็บ	seven satang piece	ချုပ်
951	ฟัน (ด้วยของมีคม)		ခုတ်
952	ครั้ง		ချက်
953	เสียงเท้า	a twenty -five	ခြေသံ
954	ทันสมัย	one	ခေတ်မှီ
955	รองเท้าแตะ	two	ခြေညှပ်ဖိနပ်
956	แข็งแรง	three	ခိုက်
957	กะลา	four	ခွက်

958	คงทน	five	ခိုင်မာ
959	เปลือกหอย	six	ချုခွံ
960	ประเดี๋ยวเดียว	seven	ခဏနှင့်
961	หัวหน้า	eight	ခေါင်းဆောင်
962	กระเจี๊ยบแดง	nine	ချဉ်ပေါင်
963	มะเขือ	ten	းခရမ်
964	มะเขือเทศ	eleven	ခရမ်းချဉ်
965	เหงื่อไคล	twelve	ချေဩ်
966	กระโดดโลดเต้น	twenty	းခုန်ပေါက်ပြေးလွှာ
967	เปรียบเทียบ	twenty one	းခိုင်းနှိုင်
968	หมุน	thirty	ချာ
969	ปู	thirty two	းခင်
970	เล้า	forty	ခြံ
971	ไข่ไก่ในเล้า	fifty	ခြံပေါက်ကြက်ဥ
972	พับ	sixty	ခေါက်
973	ร่างกาย	seventy	ခ-ၵကိုယ်
974	ไอ	block	းချောင်းဆို
975	เหือดแห้ง	car	းခန်
976	หลีกเลี่ยง	give medical treatment	းခိုတမ်
977	หัก	help	ခုန်မ်

978	สิงโต	copy	ခြေသ့်
979	ร่ำรวย	cross over	ချမ်းသာ
980	พาไป	interjection	းခေါ်သွာ
981	ผูก	silt up	ချည်
982	ภามี	glue	းခွန်အာ
983	คร่อม	one s deed	ခွ
984	เข้าไปใกล้	be lucky	ချည်းကပ်
985	สะคือ	worship	ချက်
986	ตะกร้อ		းခြင်းလုံ
987	อุ้งเท้า	canopy	ခြေခွင်
988	ซนเท้า	cock	ခြေဖနောင့်
989	หลังเท้า	adore	းခြေဖမို
990	ปลายเท้า	carry	ခြေဖျားထောက်
991	วงตะกร้อ	constant	ခြင်းရိုင်းခတ်
992	นักท่องเที่ยว , ผู้โดยสาร	arched	ခရီးသည်
993	สะดวก , สบาย	grow up	ချောင်လည်
994	รัก	grassland	ချစ်
995	คลั่ง	use	ချိပ်
996	หวานซึ้ง	ait	ချိ မြ မြ
997	หลอก	kitchen garden	ချောက်

998	รู้สึก	drop out	းခံစာ
999	แยก	blouse	းခွဲခြာ
1000	คู่ครอง	dance	းခင်မွန်
1001	ขูด	scrape	ခြစ်
1002	ชิงช้า	swing	းချာ
1003	กระพร้อ		ခွက်ငယ်
1004	ประมาณ	about	ခန့်
1005	สรุป	abstract	ချုပ်ပြောရယင်
1006	เขตสัมปทาน	concession	ချထားသောမြေကွက်
1007	กันดาร	lack of	ခေါင်
1008	บังคับ	beat into	းခြိမ်းခြောက်အပိုင်စီ
1009	ถ้ำ	cave	ုဂ
1010	โคตมะ		ဂေါတမ
1011	เกียรติ	honor	ုတ်ဏာဂ
1012	เอาใจใส่	extroverted	စိုက်ရဂ
1013	ยิมนาสติก	gymnastic	ှမ်းဘာဂ
1014	ครุฑ	Garuda	ုဋ္ဌဂ
1015	คนญี่ปุ่น	Nipponese	ျပန်နိုင်ံဂ
1016	ภูมิใจ	proud	ူဏာ်ယူဂ
1017	แป้ง	new-milled flour	ှုတဘက်ဂ

1018	จมูก	nose	ယါန
1019	ฆราวาส	laical	ယရာဝါသ
1020	ความรู้สึกทางกลิ่น	smell	ဃါနယတန
1021	เป็นคำย่อ ฉัน , ข้า		c
1022	คำสรรพนาม ชาย , หญิง		ငါ
1023	ปลา , ห้า (ตัวเลข)	fish	ငါး
1024	นกเขาเขียว	turtle	ဋ္ဌ
1025	ปล่อยปลา	to free	ဋ္ဌါလှတ်
1026	เศษสตางค์	part of a baht; small change	ငွေအင်္ဂက
1027	ภัยที่เกิดจากนก	menace	ငှက်ဘေ
1028	ซึม	glassy	ငေးမှိုင်
1029	น้ำมันตับปลา	cod-liver oil	ငါးကြီးဆီ
1030	สงบนิ่ง	equable	ငြိမ်
1031	นกคุ้ม	quail	ငုံ
1032	ง่วง	sleepy	ငိုက်
1033	เรา	We	ငါတို့
1034	พริกสด	chili	ငရုတ်စိမ်
1035	พริกไทย	pepper	ငရုတ်ကောင်
1036	โป๊ะ	pontoon	ငါးဖမ်းဆည်
1037	ปลาจาระเม็ด	pomfret	ငါးမုတ်

1038	เช็ด	to wipe , to scrub , to rub, to clean	ၵဲ
1039	ปลาตะเพียนใหญ่	Hami ton carp	ꩡငါးကြာင်
1040	เพื่อน	friend	ꩡငယ်ချင်
1041	ทะเลาะ	refuse to obey	ငြင်းခုံ
1042	ปลาหมอ	climbing perch	ငါးပြေမ
1043	ปลาช่อน	snake-head fish	ငါးရပ်
1044	ปลาหมึก	cuttlefish	ꩡငါးပြည်ကြီ
1045	5 อัน	five piece	ငါးခု
1046	เบ็ด	fishing rod	ငါးမျှား
1047	ขังปลาไว้	to cage a fish	ꩡငါးလှောင်ထာ
1048	จับปลา	catch fish	ငါးဖမ်
1049	เช่า	rent	ꩡငှါ
1050	บ้านเช่า	rented house	ငှါးအိမ်
1051	เช่าบ้าน	rent a house	ꩡအိမ်ငှါ
1052	เค็ม	salty	ငန်
1053	ห่าน	goose	ꩡငှန်
1054	ปั่น (ฝ้าย)	churn into	ငင်
1055	อย่างสงบ	quiescently	ငြိမ်သက်စွာ
1056	เจ้าของห่าน		ငန်းရှင်
1057	สงบสุข	peaceable	ꩡငြိမ်

1058	สันติภาพ	peace	်ရငြိမ်းချမ်းေ
1059	เงินค่าปรับ	indemnity	ငွေဒဏ်
1060	เงินเฟ้อ	inflation	ငွေဖောင်းပွပ၍
1061	เงินทุน	treasury	်ငွေရင်
1062	ไฟแนนท์	finance	%ငွေကြြးကုက်
1063	แผ่นดินไหว	earthquake	ငလျင်လှုပ်
1064	ฉลอง 25 ปี	twenty-five year aniversary	ငွေရတု
1065	หนังสือ	book	စာ
1066	นกกระจอก	sparrow	်စာကလေ
1067	อยากให้	want in	စေချင်
1068	กิน	eat	်စာ
1069	เรียง	line up	စီ
1070	ขี่	ridden	်စီ
1071	ตัวอย่าง	example	စံပြ
1072	ความเป็นอยู่	livelihood	်ရစား၀တ်နေေ
1073	รวม	add	စု
1074	แลบ (ลิ้น)	bulge	့ူ
1075	ตำ	pestle	်စူ
1076	เมล็ด	seed	ေစ့
1077	ส่ง	assign	ေစ

1078	เหนียว	gluey	းစေ
1079	หยุด	break	စဲ
1080	ใช้กับ	apply to	စော်
1081	เช้า	morning	စော
1082	ตามหลัง	drop astern	စို့
1083	เรียงไว้	keyword in context	းစီထာ
1084	บัญชี , รายการ	account	းစာရင်
1085	รวบรวม	aggregate	းစုပေါင်းထာ
1086	ลองดู	take a risk	စမ်းကြည့်
1087	เตรียม	arrange	စီစဉ်
1088	ห้องเครื่องจักร	workshop	စက်အို
1089	ความคิดใหม่ ๆ	live in the present	စက်ကူးသစ်
1090	ช่างเครื่องจักร	engine oil	စက်ဆီ
1091	เพชร	diamond	စိန်ကျောက်
1092	คิด	think	းစဉ်းစာ
1093	สงคราม	war	စစ်ပွဲ
1094	พรมแดน	borderland	းစည်
1095	หนังสือ	book	စာ
1096	มาตรฐาน	standard	စံ
1097	กลอง	big drum	စည်

1098	ชุมนุม	crowd	ဒုဝေ
1099	สุด	be over and above	ဒွန်
1100	เยอะ	be alive with a teeming crowd	ဒည်ကာ
1101	โต๊ะ	table	စာပွဲ
1102	สละ	throw away	စွန့်
1103	เหยียดหยาม	insult	ဒစော်ကာ
1104	เขียว	green	ဒစိမ်
1105	พร้อม	be sufficient	စုံ
1106	ขวัญ	morale	စိတ်ဓါတ်
1107	ครบ	due	ဝေ့
1108	นกเหยี่ยว	hawk	စွန်
1109	ผ้าห่ม	blanket	စောင်
1110	คณะทำงาน	organize	စီမံခန့်
1111	ผ้าลูกไม้	lacework	ဗျာ
1112	ตลาด	market	ဒဈေ
1113	รั่ว	leak	စိမ်
1114	ดอกมะลิ	jasmine	ဒစပယ်ပန်
1115	พืชที่กินได้		စားပင်
1116	โรงงาน	factory	စက်ရုံ
1117	กระดาษ	paper	စက္ကူ

1118	ว่าว	kite	စွန်
1119	ราคาถูก	low price,save,economic	ဈေးချို
1120	ดูแล	protect	စောင့်ရောက်
1121	ตาย	die	စုတ္
1122	หิ้ง	platform	စင်
1123	มัด , พัน	fence	းစည်
1124	พิสูจน์	make enquiries	းစုံစမ်
1125	สะสม	be in a mass	စုဆောင်
1126	เครื่อง	controlling power	စက်
1127	พลทหาร	private	းစစ်သားကလေ
1128	แตกแยก	disintegrate	စိတ်ဝမ်းကွဲ
1129	หรือ	ask	စေ
1130	คอย	bide	စောင့်
1131	กำลังใจ	morale	းစိတ်အာ
1132	วางแผน	concoct	စီမံ
1133	พลังงาน	energy	စွမ်းအင်
1134	บรรทม	sleep	စက်တော်ခေါ်သည်
1135	รวบรวม	aggregate	းစုရုံ
1136	ระบบ	system	စနစ်
1137	เนื้อความ	attachment	းစာသာ

1138	ตัดสิน	arbitrate	စီရင်
1139	ถ่าย (อุจจาระ)	faeces	စွန့်
1140	เปียก	dank	စို
1141	พิณ	spinet	စောင်းကောက်
1142	เริ่ม	begin	စတင်
1143	สำนวน	idiom	းစကားထူ
1144	ตาราง	table	းစတုရန်
1145	แกลบ	husk	စပါးခွံ
1146	บรรจุ	carry	စွက်
1147	หิว	hungry	ဆာ
1148	เกลือ	salt	းဆာ
1149	น้ำมัน	oil	ဆီ
1150	ห้าม	ban	းဆီ
1151	พุทรา	monkey apple	းဆီးသီ
1152	เดือด	boil	ဆူ
1153	หนาม	prickle	းဆူ
1154	ยา	medicine	းဆေ
1155	ค่า	worth	ဆဲ
1156	เร็ว	suddenly	ဆော
1157	บัดกรี	solder	ဆော်

1158	ผม	hair	ဆံ
1159	จุ (ปริมาณ)	bulk	ဆုံ့
1160	ตักบาตร	put food into a monk bowl	ဆွမ်းလောင်း
1161	ช่วยเหลือ	rescue	ဆယ်
1162	ซน	mischievous	ဆော့
1163	ดึง	abstract	ဆွဲ
1164	รักษา	adiminister	ဆေးကု
1165	สถานีอนามัย	health station	ဆေးပေးဌာန
1166	ช้าง	elephant	ဆင်
1167	ปฏิบัติ	carry with one	ဆောင်
1168	มวยผม	coil hair or tie hair in a knot	ဆံထုံး
1169	ยา	potion	ဆေးဝါး
1170	คั่งค้าง	be overdue	ဆိုင်းတွ
1171	กระตุก	flick	ဆတ်
1172	จุ่ม	dip in	ဆွတ်
1173	ปลายข้าว	grist	ဆန်ကွဲ
1174	ขออวยพร	sanctify	ဆုတောင်း
1175	เหยียด	lay over	ဆန့်
1176	บทความ	article	ဆောင်းပါး
1177	ซัง	the serew - pine	ဆပ်သွားဖူး

1178	จาวมะพร้าว	some tiime found inside a coconut	ဆံခြည်ထွေး
1179	ผงซักฟอก	soap powder	ဆပ်ပြာမှုန့်
1180	แบบก่อสร้าง	design	ဆောက်လုပ်ထားပုံ
1181	ฤดูหนาว	cold season	ဆောင့်
1182	ทำ	do	ဆောင်
1183	หมอ	docter	ဆရာဝန်
1184	ปลูกสร้าง	buildings	ဆောက်လုပ်
1185	ผักสลัด	salad greens	ဆလတ်ရွက်
1186	โรคเบาหวาน	diabetes	ဆီးချိုရောဂါ
1187	ยากจน	poor	ဆင်းရဲ
1188	สั่งสอน	instruct	ဆုံးမ
1189	กล่าว	speech	ဆိုရိုး
1190	ประจำ	be stationed	ဆိုင်ရာ
1191	ตัดสินใจ	conclude	ဆုံးဖြတ်
1192	สอนให้	educable	ဆုံးမပေး
1193	สีข้าว	mill paddy or rice	ဆန်ကြိတ်
1194	ชะอม	kind of vegetable with a strong odour	ဆူးပူပ်
1195	กาง (กางร่ม)	cover one head	ဆောင်း
1196	จิก	peck	ဆိတ်ပေါက်
1197	แจกัน	goglet	တကောင်း

1198	ไวโอลิน	violin	တယော
1199	การต่อต้าน	anti	ဆန့်ကျင်ရေး
1200	สัปทน	that part of an umbrella	ထီးခေါင်း
1201	เขื่อน	dam	ဆည်
1202	คลองชลประทาน	canal	ဆည်မြောင်း
1203	น้ำสบู่	suds	ဆပ်ပြာရည်
1204	คำตัดสิน	decision	ဆုံးဖြတ်ချက်
1205	โรงสีข้าว	rice mill	ဆန်စက်
1206	ตะแกรง	grate	ဆန်ကာ
1207	ร่อนลง	volplane	ဆန်ခါချ
1208	คราบน้ำมัน	slick	ဆီချေး
1209	แพรวพราว	gorgeous	ဆန်းကြယ်
1210	อาหาร	food	ဆန်ရေစပါး
1211	พิจารณา	allow for	ဆင်ခြင်း
1212	ดอกโสน		ဇွန်ပန်း
1213	ความพยายาม	temptation	ဇွဲ
1214	ภรรยา	wife	ဇနီး
1215	ดื้อดึง	contumacious	ဇွတ်တရွတ်
1216	เป็นอย่างนี้	.That's how it goes	ဇာတိပဲ
1217	ใหม่	new	ဇော

1218	จริง ๆ	indeed	ကေနဲ
1219	ความเป็นชาตินิยม	patriotism	ဇာတိသွေးဇာတိမာန်
1220	ความเย็น	coldness	မြ
1221	เอเคอร์		က
1222	กะละมัง	basin	ဇလုံ
1223	พุทรา		းဒီးဒီ
1224	ราง	groove	းဒီးလာ
1225	มะขามป้อม		းဒီးဥပုသီ
1226	ชมพู่มะเหมี่ยว	the rose apple	ဇမ္ဗူးသပြေ
1227	แหล่งกำเนิด	genesis	ဇာစ်မြစ်
1228	ราคาแพง	expensive	းစျေးကြီ
1229	ราคาถูก	cheapness	စျေးတော်
1230	กลางคืน	night	ည
1231	ขวา	right	ညာ
1232	แนะนำ	acquaint	ဆွန်ပြ
1233	น้องชาย	younger brother	ညီ
1234	น้องสาว	younger sister	ညီမ
1235	เสมอ	always	ညီညာ
1236	อย่างละเอียด	finely	ညက်ညက်
1237	เท่ากัน	commeasure	ညီမျ

1238	สกปรก	dirty	ညစ်ပေ
1239	เมื่อย	feel stiff	ညောင်း
1240	ทวีปยุโรป	Europe	ဥပရောတိုက်
1241	เบื้องต้น	elementary	ဦးစွာ
1242	ต้นโพธิ์	pipal	ညောင်ပင်
1243	ปัญญา	Intellect	ညဏ်
1244	ประสาน	harmonize	ညှိနှိုင်
1245	ระยะ	carry	တာ
1246	ห้าม	block	းတာ
1247	1 ศอก	one cubit	တတောင်
1248	ผลัก	crowd	းတွန်
1249	ตรง	rush	တိ
1250	ไส้เดือน	earthworm	တိ
1251	สั้น	abbreviated	တို
1252	ขึง	strain	းတန်
1253	ดีด	chord	တီး
1254	ศาลาพัก	pavilion	တဉ်း
1255	ปลอม	artificial	တု
1256	เหมือน	take after	တူ
1257	ทั้งหลาย	all	တွေ

1258	ขุด	dig for	တူး
1259	สกปรก	dirty	တေ
1260	เพลง	song	တေး
1261	บนยอดเขา	mountaintop	တောင်ထိပ်ပေါ်မှာ
1262	คิด	be thoughtless	တွေးတော
1263	กระท่อม	cottage	တဲကုပ်
1264	ไม้เท้า	walking stick	တောင်ဒေး
1265	นักปฏิวัติ	revolutionist	တော်လှန်ရေးစမား
1266	วัดป่า	a wat, a temple, a church	တောရကျောင်း
1267	ข้อเสนอ	offer	တင်ပြချက်
1268	ตลอด	across	တလျှောက်လုံး
1269	เสนอ	across	တင်ပြ
1270	แสดงอาการ	act up	တန်း
1271	ตรงไป	direct	တောက်လျှောက်
1272	ลาน	garth	တလင်း
1273	ที่	that	တဲ့
1274	บรรทุก	put or place	တင်
1275	พัก	lie in	တယ်
1276	กระดก	steeve	တော့
1277	ขุด	dig	တူးဖေါ်

1278	ฟังธรรม	listen to sermon	တရားနား
1279	ขัน	crow	တွန်
1280	เกาะ	cling onto	တွယ်
1281	อย่างใดอย่างหนึ่ง	one thing	တခုခု
1282	หน้าที่	agency	တာဝန်
1283	ติดตามดู	spy on	တွယ်တာစရာ
1284	กลุ่มหนึ่ง	cater to so	တသင်း
1285	เสา	column	တိုင်
1286	ประดิษฐ์	create	တီထွင်
1287	วัด	audit	တိုင်
1288	ราว	so	တန်း
1289	ชายคา	eaves	တံစက်မြိတ်
1290	ประตู	door	တံခါး
1291	กำแพง	fence	တံတိုင်း
1292	อย่างอึกทึก	noisy	တအုန်းအုန်း
1293	เนินเขา	hill	တောင်ကုန်း
1294	นักเรียน	pupil	တပည့်
1295	ซึ่ง	which	တရော
1296	ทั่ว	about	တခုလုံး
1297	เพิ่มพูน	augment	တိုးတက်

ภาคผนวก ค

ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] P. Limwong , C. Benjangkaprasert and O.sangaroon "Intelligent Searching with Multi-Display by Using MATLAB", SICE-ICASE International Joint Conference 2006, Bexco, Busan, Korea, pp. 3166-3168, Oct. 18-21, 2006.

ประวัติผู้เขียน

พันโท พีรเดช ลิ้มวงศ์ เกิดเมื่อวันที่ 23 กรกฎาคม 2502 ที่ อำเภอพิชัย จังหวัด อุตรดิตถ์ สำเร็จการศึกษาระดับปริญญาตรี จาก ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สาขาวิชาเทคโนโลยีโทรทัศน์ ปีการศึกษา 2527 และ สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ ปีการศึกษา 2531 เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2545 ปัจจุบันทำงานที่ ศูนย์รักษาความปลอดภัย กองบัญชาการทหารสูงสุด