สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมเพื่อลดขนาดข้อมูลโดยวิธี ADAPTIVE COMPRESSION

DEVELOPMENT DATA COMPRESSION WITH ADAPTIVE COMPRESSION

T 047353

ชัยชนะ วัยวุฒิเกียรติ
บัณฑิต รังสีรัตนากุล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 47353
วัน, เดือน, ปี 30 มิ.ย. 2546

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

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPMENT DATA COMPRESSION WITH ADAPTIVE COMPRESSION

CHAICHANA WAIWUTTIKIAT

BANDIT RUNGSRERATANAKUL

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาพิเศษเรื่อง	การพัฒนาโปรแกรมเพื่อลดขนาดข้อมูล โดยวิธี ADAPTIVE COMPRESSION DEVELOPMENT DATA COMPREESION WITH ADAPTIVE COMPRESSION
ชื่อนักศึกษา	นายชัยชนะ วัยวุฒิเกียรติ 42050012 นายบัณฑิต รังสีรัตนากุล 42050026
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา	คณิตศาสตร์ประยุกต์
อาจารย์ที่ปรึกษา	ผศ.สุนทร สุชาติเวชภูมิ วิสันต์ ตั้งวงษ์เจริญ

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิชาคณิตศาสตร์ประยุกต์ ปีการศึกษา 2545

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	รศ.ภัคนี ชิตสกุล	
กรรมการ	ผศ.กฤษฎา ไตรสุรัตน์	
กรรมการและอาจารย์ที่ปรึกษา	ผศ.สุนทร สุชาติเวชภูมิ	
กรรมการและอาจารย์ที่ปรึกษา	อ.วิสันต์ ตั้งวงษ์เจริญ	

(ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาพิเศษเรื่อง การพัฒนาโปรแกรมเพื่อลดขนาดข้อมูลโดยวิธี ADAPTIVE COMPRESSION

ชื่อนักศึกษา นายชัยชนะ วัยวุฒิเกียรติ 42050012
นายบัณฑิต รังสีรัตนากุล 42050026

ปริญญา วิทยาศาสตรบัณฑิต
ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา คณิตศาสตร์ประยุกต์
ปีการศึกษา 2545

อาจารย์ที่ปรึกษา ผศ.สุนทร สุชาติเวชภูมิ
วิสันต์ ตั้งวงษ์เจริญ

บทคัดย่อ

Adaptive Huffman เป็นวิธีลดขนาดข้อมูลโดยอาศัยทฤษฎีทางคณิตศาสตร์เรื่องความน่าจะเป็นและทฤษฎีแผนภูมิต้นไม้มาช่วยในการแปลงรหัสแอสกีให้เป็นโค้ดที่มีจำนวนบิตน้อยลงทำให้ข้อมูลมีขนาดเล็กลง เพื่อความสะดวกในการจัดเก็บและรับส่งข้อมูลผ่านเครือข่าย ซึ่งจากการศึกษาจะพบว่าวิธี Adaptive Huffman เหมาะสมที่จะใช้ลดขนาดข้อมูลชนิดตัวอักษร แต่ถ้านำไปใช้ลดขนาดไฟล์ที่ถูกบีบอัดมาแล้วเช่น JPG และ GIF เป็นต้นจะได้ประสิทธิภาพที่ไม่น่าพอใจ ยกตัวอย่างไฟล์ JPG เป็นไฟล์กราฟฟิกที่ได้ถูกบีบอัดมาก่อนแล้ว โดยที่ข้อมูลในไฟล์ JPG จะถูกแบ่งย่อยออกมาและจัดเก็บในรูปเมทริกซ์จัตุรัส จึงทำให้ข้อมูลมีโอกาสซ้ำกันน้อยมาก ทำให้ไฟล์ที่ได้มีขนาดเกือบเท่ากับไฟล์ตัวเดิม แสดงให้เห็นว่าวิธี Adaptive Huffman นั้นยังไม่สามารถจัดเรียงโค้ดหลังจากทำการลดขนาดได้ดีเพียงพอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	DEVELOPMENT DATA COMPRESSION WITH ADAPTIVE COMPRESSION
Students	Mr. Chaichana Waiwuttikiat 42050012
	Mr. Bandit Rungsrerattanakul 42050026
Degree	Bachelor of science
Department	Mathematics and Computer Science Faculty of science
Programme	Applied Mathematics
Academic Year	2002
Special Project Advisor	Asst.Prof. Sunthorn Suchatvejapoom
	Wisan Tangwongcharone

ABSTRACT

The purpose of this project is to present the development of data compression with adaptive compression. Adaptive Huffman is a method to minimize the size of data with application of probability of mathematics theory and tree diagram to convert ASCII code into the code of less bits. By this mean, minimized data will be achieved with the size of more facilitating for filing and data communication through networks. This study observed that Adaptive Huffman method is suitable for minimizing size of text data. The method gave unsatisfied results for already-compressed files such as JPG or GIF. The reason is that such kind of files as JPG contains apportioned data being filed in the form of square matrix. This leads to such very little possibility of data repetition that the new file got has nearly same size as its original. Summarily, it might be concluded that Adaptive Huffman method could not effectively the codes of data that has been undergone a size minimization.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องการพัฒนาโปรแกรมเพื่อลดขนาดข้อมูลโดยวิธี Adaptive Compression สามารถสำเร็จลุล่วงไปด้วยดีคณะผู้จัดทำต้องขอขอบพระคุณ

ผู้ช่วยศาสตราจารย์สุนทร สุชาติเวชภูมิ

อาจารย์วิสันต์ ตั้งวงษ์เจริญ

อาจารย์ที่ปรึกษาปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาในการแก้ปัญหาต่างๆ รวมทั้งเป็นผู้ตรวจสอบความถูกต้องอขงปัญหาพิเศษฉบับนี้

ขอขอบพระคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ ที่ให้ความสะดวกในการใช้ห้องปฏิบัติการคอมพิวเตอร์ และในการจัดเบิกอุปกรณ์ต่างๆ ที่ใช้จัดทำปัญหาพิเศษ

ที่สำคัญที่สุดคณะผู้จัดทำต้องขอขอบพระคุณ บิดา มารดา ที่ให้ความสนับสนุนทางด้านการเรียนและกำลังใจ จนการทำปัญหาพิเศษนี้สำเร็จลงด้วยดี รวมทั้งเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เกี่ยวกับปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VII

สารบัญรูป.....IX

บทที่ 1 บทนำ.....1

1.1 ความสำคัญและที่มาของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 สมมติฐานของการศึกษา.....2

1.4 ขอบเขตของปัญหา.....2

1.5 ขั้นตอนการศึกษา.....2

1.6 ข้อจำกัดการศึกษา.....3

บทที่ 2 ความรู้และทฤษฎีที่เกี่ยวข้อง.....4

2.1 ประเภทของการลดขนาดข้อมูล.....4

2.1.1 วิธี Lossy Data Compression.....4

2.1.2 วิธี Lossless Data Compression.....4

2.2 นิยามของการลดขนาดข้อมูล.....4

2.2.1 การกำหนดโค้ด (Coding).....7

2.2.2 การทำโมเดล (Modeling).....7

2.3 ทฤษฎีแผนภูมิต้นไม้.....7

2.4 ทฤษฎีความน่าจะเป็นในการลดขนาดข้อมูล.....10

2.5 การลดขนาดโดยแนวคิดทางคณิตศาสตร์.....11

2.5.1 วิธี Huffman Coding.....11

2.5.2 วิธี Adaptive Huffman Coding.....15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินการวิจัย.....24

3.1 วิธีการดำเนินการวิจัย.....24

3.2 ระบบงาน.....24

3.2.1 ลักษณะของข้อมูล (Input) 24

3.2.2 ส่วยวิเคราะห์และประมวลผลด้วยวิธี Adaptive Huffman Coding.....24

3.2.3 ส่วนแสดงผล (Output).....24

3.3 ขั้นตอนการดำเนินงาน.....24

3.4 ขั้นตอนการทำงานของโปรแกรม Adaptive Huffman.....25

บทที่ 4 ผลการทดลอง.....**27**

4.1 ขั้นตอนการทดลอง.....27

4.2 ผลการทดลองการลดขนาดข้อมูลชนิดต่างๆ ด้วย Adaptive Huffman.....28

4.2.1 ผลการทดลองลดขนาดข้อมูลชนิด DOC ด้วยวิธี Adaptive Huffman.....28

4.2.2 ผลการทดลองลดขนาดข้อมูลชนิด TXT ด้วยวิธี Adaptive Huffman 29

4.2.3 ผลการทดลองลดขนาดข้อมูลชนิด JPG ด้วยวิธี Adaptive Huffman.....30

4.2.4 ผลการทดลองลดขนาดข้อมูลชนิด GIF ด้วยวิธี Adaptive Huffman.....31

4.2.5 ผลการทดลองลดขนาดข้อมูลชนิด WAV ด้วยวิธี Adaptive Huffman.....32

4.2.6 ผลการทดลองลดขนาดข้อมูลชนิด MP3 ด้วยวิธี Adaptive Huffman.....33

4.2.7 ผลการทดลองลดขนาดข้อมูลชนิด EXE ด้วยวิธี Adaptive Huffma 34

4.2.8 ผลการทดลองลดขนาดข้อมูลชนิด DAT ด้วยวิธี Adaptive Huffman.....35

4.2.9 ผลการทดลองลดขนาดข้อมูลชนิด WMV ด้วยวิธี Adaptive Huffman 36

4.3 ผลการลดขนาดครั้งที่ 2 หลังจากลดด้วย Adaptive Huffman.....37

4.3.1 ผลการลดขนาดข้อมูลชนิด DOC ครั้งที่ 2.....37

4.3.2 ผลการลดขนาดข้อมูลชนิด TXT ครั้งที่ 2.....38

4.3.3 ผลการลดขนาดข้อมูลชนิด JPG ครั้งที่ 2.....39

4.3.4 ผลการลดขนาดข้อมูลชนิด GIF ครั้งที่ 2.....40

4.3.5 ผลการลดขนาดข้อมูลชนิด WAV ครั้งที่ 2.....41

4.3.6 ผลการลดขนาดข้อมูลชนิด MP3 ครั้งที่ 2.....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.3.7 ผลการลดขนาดข้อมูลชนิด EXE ครั้งที่ 2.....43
4.3.8 ผลการลดขนาดข้อมูลชนิด DAT ครั้งที่ 2.....44
4.3.9 ผลการลดขนาดข้อมูลชนิด WMV ครั้งที่ 2 45
4.4 ผลการทดลองด้วยวิธีต่างๆ เปรียบเทียบกับ Adaptive Huffman.....46
4.4.1 ผลการทดลองการลดขนาดข้อมูลชนิด DOC ด้วยวิธีต่างๆ.....46
4.4.2 ผลการทดลองการลดขนาดข้อมูลชนิด TXT ด้วยวิธีต่างๆ 47
4.4.3 ผลการทดลองการลดขนาดข้อมูลชนิด JPG ด้วยวิธีต่างๆ.....48
4.4.4 ผลการทดลองการลดขนาดข้อมูลชนิด GIF ด้วยวิธีต่างๆ.....49
4.4.5 ผลการทดลองการลดขนาดข้อมูลชนิด WAV ด้วยวิธีต่างๆ.....50
4.4.6 ผลการทดลองการลดขนาดข้อมูลชนิด MP3 ด้วยวิธีต่างๆ.....51
4.4.7 ผลการทดลองการลดขนาดข้อมูลชนิด EXE ด้วยวิธีต่างๆ.....52
4.4.8 ผลการทดลองการลดขนาดข้อมูลชนิด DAT ด้วยวิธีต่างๆ 53
4.4.9 ผลการทดลองการลดขนาดข้อมูลชนิด WMV ด้วยวิธีต่างๆ.....54

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....55
5.1 สรุปผลการทดลอง.....55
5.2 สรุปผลการทดลองเปรียบเทียบกับวิธีตัวอื่นๆ.....57
5.3 สรุปผลการลดขนาดครั้งที่ 2 หลังจากลดขนาดครั้งแรกด้วย Adaptive Huffman.....59
5.4 ข้อจำกัดการใช้โปรแกรม.....61
5.5 ข้อเสนอแนะ 61

บรรณานุกรม.....62
ภาคผนวก ก ตัวอย่างการใช้โปรแกรม.....63
ภาคผนวก ข อธิบายการทำงานของโปรแกรม.....69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางตัวอย่างของการหาค่าความน่าจะเป็นจากตัวอย่างที่ 2.2 11
2.2 ตารางแสดงจำนวนความถี่ของตัวอักษรในข้อมูลชุดหนึ่งจากตัวอย่างที่ 2.3 12
2.2 ตารางแสดงจำนวนความถี่ของตัวอักษรในข้อมูลชุดหนึ่งจากตัวอย่างที่ 2.3 (ต่อ) 13
2.3 ตารางแสดงโค้ดที่ได้จากวิธี Huffman 15
2.4 ตารางแสดงโค้ดที่ได้จากวิธี Adaptive Huffman 22
2.5 ตารางแสดงการเปรียบเทียบจำนวนโค้ดที่ได้ 22
4.2.1 แสดงผลการลดขนาดไฟล์ DOC ด้วยวิธี Adaptive Huffman 28
4.2.2 แสดงผลการลดขนาดไฟล์ TXT ด้วยวิธี Adaptive Huffman 29
4.2.3 แสดงผลการลดขนาดไฟล์ JPG ด้วยวิธี Adaptive Huffman 30
4.2.4 แสดงผลการลดขนาดไฟล์ GIF ด้วยวิธี Adaptive Huffman 31
4.2.5 แสดงผลการลดขนาดไฟล์ WAV ด้วยวิธี Adaptive Huffman 32
4.2.6 แสดงผลการลดขนาดไฟล์ MP3 ด้วยวิธี Adaptive Huffman 33
4.2.7 แสดงผลการลดขนาดไฟล์ EXE ด้วยวิธี Adaptive Huffman 34
4.2.8 แสดงผลการลดขนาดไฟล์ DAT ด้วยวิธี Adaptive Huffman 35
4.2.9 แสดงผลการลดขนาดไฟล์ WMV ด้วยวิธี Adaptive Huffman 36
4.3.1 แสดงผลการลดขนาดข้อมูลชนิด DOC ครั้งที่ 2 37
4.3.2 แสดงผลการลดขนาดข้อมูลชนิด TXT ครั้งที่ 2 38
4.3.3 แสดงผลการลดขนาดข้อมูลชนิด JPG ครั้งที่ 2 39
4.3.4 แสดงผลการลดขนาดข้อมูลชนิด GIF ครั้งที่ 2 40
4.3.5 แสดงผลการลดขนาดข้อมูลชนิด WAV ครั้งที่ 2 41
4.3.6 แสดงผลการลดขนาดข้อมูลชนิด MP3 ครั้งที่ 2 42
4.3.7 แสดงผลการลดขนาดข้อมูลชนิด EXE ครั้งที่ 2 43
4.3.8 แสดงผลการลดขนาดข้อมูลชนิด DAT ครั้งที่ 2 44
4.3.9 แสดงผลการลดขนาดข้อมูลชนิด WMV ครั้งที่ 2 45
4.4.1 แสดงผลการลดขนาดไฟล์ DOC ด้วยวิธีต่างๆ 46
4.4.2 แสดงผลการลดขนาดไฟล์ TXT ด้วยวิธีต่างๆ 47
4.4.3 แสดงผลการลดขนาดไฟล์ JPG ด้วยวิธีต่างๆ 48
4.4.4 แสดงผลการลดขนาดไฟล์ GIF ด้วยวิธีต่างๆ 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.4.5 แสดงผลการลดขนาดไฟล์ WAV ด้วยวิธีต่างๆ.....50
4.4.6 แสดงผลการลดขนาดไฟล์ MP3 ด้วยวิธีต่างๆ.....51
4.4.7 แสดงผลการลดขนาดไฟล์ EXE ด้วยวิธีต่างๆ.....52
4.4.8 แสดงผลการลดขนาดไฟล์ DAT ด้วยวิธีต่างๆ.....53
4.4.9 แสดงผลการลดขนาดไฟล์ WMV ด้วยวิธีต่างๆ.....54
5.1 ตารางแสดงผลการลดขนาดไฟล์เฉลี่ยประเภทต่างๆด้วยวิธี Adaptive Huffman 55
5.2 ตารางแสดงการเปรียบเทียบผลการลดขนาดครั้งที่ 2 หลังจากลดด้วย Adaptive Huffman.....57
5.3 ตารางแสดงผลการทดลองลดขนาดข้อมูลเฉลี่ยเปรียบเทียบกันด้วยวิธีต่างๆ.....59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 แผนผังการลดขนาดข้อมูลด้วยวิธี Adaptive Huffman 6
2.2 ตัวอย่างแผนภูมิต้นไม้ 8
2.3 แผนภูมิต้นไม้ในตัวอย่างที่ 2.1 9
2.4 ตัวอย่างแผนภูมิต้นไม้แบบ Binary Tree 10
2.5 แผนภูมิต้นไม้จากวิธี Huffman หลังจากรอบแรก.....11
2.6 แผนภูมิต้นไม้ของวิธี Huffman หลังจากรอบที่สอง.....14
2.7 แผนภูมิต้นไม้ Huffman 15
2.8 แผนภาพแสดงการเปลี่ยนแปลงของ Tree หลังจากพบตัวอักษร D.....17
2.9 แผนภาพแสดง Tree หลังจากพบ A.....17
2.10 แผนภาพแสดงแผนภูมิต้นไม้หลังจากทำการสลับที่โหนด.....18
2.11 แผนภาพแสดงแผนภูมิต้นไม้เมื่อเพิ่ม B เข้าไป.....18
2.12 แผนภาพแสดงแผนภูมิต้นไม้หลังจากเพิ่ม C เข้าไป.....19
2.13 แผนภาพแสดงแผนภูมิต้นไม้หลังจากทำการสลับที่ครั้งล่าสุด 20
2.14 แผนภาพแสดงแผนภูมิต้นไม้เมื่ออ่านข้อมูลครบทุกตัวแล้ว 21
3.1 แผนผังแสดงขั้นตอนการลดขนาดไฟล์.....25
3.2 แผนผังแสดงขั้นตอนการขยายข้อมูลให้กลับเป็นปกติ.....26
5.1 แผนภูมิเปรียบเทียบผลการลดขนาดไฟล์ต่างๆด้วยวิธี Adaptive Huffman 56
5.2 แผนภูมิเปรียบเทียบผลการลดขนาดครั้งที่ 2 หลังจากลดขนาดด้วย Adaptive Huffman 58
5.3 แผนภูมิเปรียบเทียบประสิทธิภาพในการลดขนาดไฟล์ประเภทต่างๆ.....60
ก.1 รูปภาพแสดงการเข้าสู่ระบบ DOS.....64
ก.2 รูปภาพแสดงวิธีเข้าสู่โฟลเดอร์ที่ต้องการในระบบ DOS.....65
ก.3 รูปภาพแสดงวิธีการใช้โปรแกรม Adaptive Huffman.....66
ก.4 รูปภาพแสดงเวลาโปรแกรมทำการลดขนาดเรียบร้อยแล้ว 67
ก.5 รูปภาพแสดงไฟล์หลังจากการลดขนาดเรียบร้อยแล้ว.....68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ปัจจุบันเทคโนโลยีในด้านต่างๆมีการพัฒนาไปอย่างรวดเร็ว ทั้งด้านวิทยาการทางการแพทย์ ด้านเกษตรกรรม ด้านอุตสาหกรรม ด้านเศรษฐกิจและสังคมโดยเฉพาะอย่างยิ่งเทคโนโลยีทางด้านคอมพิวเตอร์มีการพัฒนารวดเร็วมากกว่าด้านอื่น ๆ เนื่องจากคอมพิวเตอร์เข้ามามีส่วนช่วยพัฒนาในทุก ๆ ด้านดังกล่าว ซึ่งสนับสนุนให้เกิดประสิทธิภาพในการดำเนินงานของสาขานั้น ๆ การค้นคว้าและพัฒนาในด้านระบบคอมพิวเตอร์ การโปรแกรมและการสื่อสารผ่านเครือข่ายจึงดำเนินไปอย่างต่อเนื่อง เพื่อที่จะเพิ่มประสิทธิภาพในการดำเนินงานที่เกี่ยวข้องปัจจุบันและอนาคต

ดังนั้นเมื่อมีการใช้ระบบคอมพิวเตอร์และการสื่อสารผ่านเครือข่ายที่มีการพัฒนาให้ตอบสนองความต้องการได้อย่างสูงสุดแล้ว จึงเป็นการเพิ่มประสิทธิภาพให้กับระบบงานระบบงานอัตโนมัติที่เกี่ยวข้อง ทั้งทางด้านความสะดวกรวดเร็วในการทำงาน ความถูกต้องแม่นยำรวมไปถึงการรับ-ส่งแลกเปลี่ยนข้อมูลข่าวสารที่มีความทันสมัยและทันเหตุการณ์

ปัจจัยสำคัญ ๆ ในการพัฒนาให้เกิดระบบคอมพิวเตอร์และการสื่อสารผ่านเครือข่ายที่ดีนั้นมีอยู่หลายปัจจัยและปัจจัยที่สำคัญอย่างหนึ่งได้แก่ การลดขนาดของข้อมูล เนื่องจากในการจัดเก็บข้อมูลและการส่งข้อมูลผ่านเครือข่ายนั้น ต้องการความถูกต้องและความสะดวกรวดเร็วในการนำข้อมูลไปใช้งาน ซึ่งข้อมูลส่วนมากเป็นข้อมูลที่มีขนาดใหญ่ อาจมีผลทำให้เกิดปัญหาในการจัดเก็บลงสื่อข้อมูลบางชนิดและปัญหาความล่าช้าในการส่งข้อมูลได้ ดังนั้นจึงได้มีการคิดวิธีที่ช่วยลดขนาดข้อมูล เพื่อช่วยในการแก้ปัญหาดังกล่าว

วิธีการลดขนาดของข้อมูลนั้นมีด้วยกันหลายวิธี โดยวิธีที่ใช้กันในปัจจุบันส่วนใหญ่เป็นวิธีที่ได้มีการพัฒนามาจากวิธีการเดิมที่มีอยู่ก่อนแล้ว เช่น **Arithmetic Coding** และ **Adaptive Huffman** ได้ถูกพัฒนามาจากวิธีพื้นฐาน **Huffman Coding** ส่วน **LZ77** และ **LZ78** ได้ถูกพัฒนามาจาก **LZW** เป็นต้นดังนั้นการศึกษาวิธีการพื้นฐานของการลดขนาดของข้อมูลจึงมีความสำคัญ เพราะช่วยให้เกิดความเข้าใจในแนวคิดดั้งเดิมของวิธีนั้น ๆ และมีส่วนในการค้นคว้าพัฒนาวิธีการใหม่ ๆ ต่อไป

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อศึกษาวิธีการลดขนาดของข้อมูล ซึ่งส่วนใหญ่ข้อมูลจะมีอยู่เป็นจำนวนมากให้สามารถจัดเก็บลงอุปกรณ์เก็บข้อมูลบางชนิดที่มีขนาดความจุจำกัดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.2 เพื่อศึกษาประโยชน์ของการลดขนาดของข้อมูล ในด้านการส่งข้อมูลผ่านเครือข่ายที่มีขนาดแบนวิทธ์**(Bandwidth)**จำกัด

1.2.3 เพื่อศึกษาประโยชน์ในด้านต่าง ๆ เช่นการประหยัดค่าใช้จ่ายของการจัดหาเนื้อที่จัดเก็บหรือการจัดหาอุปกร์การจัดเก็บข้อมูล

1.2.4 เพื่อศึกษาความรู้เกี่ยวกับคณิตศาสตร์ประยุกต์ที่นำมาใช้งานกับระบบคอมพิวเตอร์

1.2.5 เพื่อประโยชน์ในการนำความรู้ที่เกี่ยวกับการลดขนาดข้อมูลไปใช้งานอย่างมีประสิทธิภาพ

1.2.6 เพื่อศึกษาหาวิธีที่จะลดขนาดข้อมูลได้อย่างมีประสิทธิภาพมากกว่าวิธี **Huffman**

1.2.7 เพื่อศึกษาแนวทางในการนำความรู้พื้นฐานในเรื่องการลดขนาดข้อมูลนำไปค้นคว้าและพัฒนาปรับปรุงเป็นวิธีการใหม่ ๆ ต่อไป

1.3 สมมุติฐานของการศึกษา

1.3.1 การลดขนาดข้อมูลแบบวิธี **Adaptive Huffman Coding** ช่วยให้ประหยัดพื้นที่และทรัพยากรในการจัดเก็บข้อมูล โดยจะใช้ได้ดีกับข้อมูลแบบ ชนิดตัวอักษร

1.3.2 ทำการลดขนาดข้อมูลก่อนที่จะส่งผ่านข้อมูลทางเครือข่าย จะช่วยเพิ่มประสิทธิภาพในการส่งผ่านข้อมูล โดยช่วยลดปัญหาในเรื่องการสูญหายของข้อมูลและยังช่วยประหยัดเวลาในการจัดส่งข้อมูล

1.3.3 ได้ตัวอย่างโปรแกรมที่ทำการลดขนาดข้อมูลให้มีขนาดเล็กลง เพื่อทำการจัดเก็บลงอุปกรณ์เก็บข้อมูล และสามารถขยายข้อมูลที่ได้ทำการลดขนาดของข้อมูลแล้วได้อย่างมีประสิทธิภาพมากกว่าวิธีการเดิม

1.4 ขอบเขตของปัญหา

ศึกษาวิธีการทางคณิตศาสตร์ที่เกี่ยวข้องกับการลดขนาดของข้อมูล โดยเน้นที่รูปแบบวิธี **Adaptive Huffman Coding** ซึ่งเป็นการลดขนาดข้อมูลในรูปแบบข้อมูลชนิดตัวอักษร ซึ่งจัดเป็นการลดขนาดของข้อมูลชนิดที่เรียกว่า **Lossless Compression** โดยจะไม่มีการสูญเสียของข้อมูลหลังจากทำการขยายขนาดข้อมูลที่ได้ทำการลดขนาดแล้ว ที่ทำการศึกษาวิธีการลดขนาดข้อมูลด้วยวิธีนี้นั้นเพราะ คณะผู้จัดทำต้องการที่จะหาวิธีที่จะลดขนาดข้อมูลได้อย่างมีประสิทธิภาพมากที่สุดเพื่อเป็นประโยชน์ในการจัดเก็บและการจัดส่งข้อมูลผ่านเครือข่ายได้รวดเร็วมากยิ่งขึ้น

1.5 ขั้นตอนการศึกษา

1.5.1 ศึกษาวิธีการและความรู้ทางคณิตศาสตร์ที่เกี่ยวข้องกับการลดขนาดของข้อมูลแบบวิธี **Adaptive Huffman Coding**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5.2 ศึกษาตัวอย่างโปรแกรมทางคอมพิวเตอร์ สำหรับการลดขนาดข้อมูลแบบวิธี **Adaptive Huffman Coding**

1.5.3 รวบรวมวิธีการที่ได้ทำการศึกษาเพื่อนำมาสร้างโปรแกรม การลดขนาดและการขยายขนาดเพื่อวัดประสิทธิภาพในการลดขนาดของข้อมูล

1.5.4 ตรวจสอบและแก้ไขโปรแกรมที่สร้างขึ้น ให้มีความถูกต้องในด้านการประมวลผล

1.5.5 ทดสอบหารูปแบบข้อมูลที่มีความเหมาะสมกับโปรแกรม

1.5.6 ทำการลดขนาดไฟล์ด้วยวิธีอื่นๆ แล้วทำการเปรียบเทียบกันวิธี **Adaptive Huffman**

1.5.7 สรุปการทำงานของโปรแกรมและทำการประเมินผล

1.6 ข้อจำกัดของการศึกษา

การศึกษาการลดขนาดของข้อมูลโดยวิธี **Adaptive Huffman Coding** เป็นการลดขนาดข้อมูลแบบข้อมูลชนิดตัวอักษรเท่านั้น ซึ่งหากมีการนำไปลดขนาดข้อมูลแบบข้อมูลชนิดอื่น ๆ อาจทำให้ผลของการลดขนาดข้อมูลมีประสิทธิภาพน้อยกว่าข้อมูลชนิดตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

2.1 ประเภทของการลดขนาดข้อมูล

2.1.1 วิธี Lossy Data Compression

Lossy Data Compression เป็นเทคนิคการลดขนาดของข้อมูลที่ยอมให้ความถูกต้องของข้อมูลสูญหายได้บ้าง ทั้งนี้ก็เพื่อความเร็วของการลดขนาดข้อมูล ซึ่งวิธีนี้ได้ถูกนำไปใช้กับงานด้านกราฟฟิกอิมเมจ **(Graphic Image)** ซึ่งเทคนิคที่ใช้สำหรับการลดขนาดของข้อมูลประเภทนี้สามารถปรับให้มีความถูกต้องได้หลายระดับ ซึ่งถ้าหากถูกกำหนดให้มีความแม่นยำสูงก็จะมีผลทำให้ขนาดไฟล์ที่ลดขนาดข้อมูลมีขนาดใหญ่ขึ้นและใช้เวลาในการประมวลผลนานขึ้นโดยเปรียบเทียบกับกรณีที่กำหนดให้มีความแม่นยำลดลง ซึ่งในปัจจุบันได้มีซอฟท์แวร์หลายประเภทที่เสนอการลดขนาดข้อมูลโดยใช้เทคนิคนี้

2.1.2 วิธี Lossless Data Compression

Lossless Data Compression เป็นเทคนิคการลดขนาดข้อมูลที่รักษาความถูกต้องของข้อมูลได้ 100% ดังนั้นไฟล์ที่ถูกลดขนาดข้อมูลโดยเทคนิคนี้จะต้องมีความถูกต้องของข้อมูลเช่นเดียวกับข้อมูลต้นฉบับทุกประการ เทคนิคส่วนใหญ่จะถูกนำไปประยุกต์ใช้กับข้อมูลแบบชนิดตัวอักษร เช่น ไฟล์ประเภทฐานข้อมูล **(database)** สเปรดชีต **(spreadsheet)** หรือข้อมูลที่ส่งตามเครื่องแฟกซ์ เป็นต้น ซึ่งไฟล์ข้อมูลประเภทนี้ล้วนมีความสำคัญทุกๆ ตัวอักษร **(byte)** ดังนั้นถ้าหากมีการสูญหายหรือเปลี่ยนแปลงในไฟล์ที่ถูกลดขนาดเพียงบิตเดียว ก็อาจจะทำให้ข้อมูลของไฟล์ที่ถูกขยายขนาดแล้วมีความคลาดเคลื่อนไป ซึ่งอาจจะไม่สามารถถูกประมวลผลได้หรืออาจทำให้ผลที่ได้ผิดพลาด เช่น โปรแกรมไฟล์

2.2 นิยามของการลดขนาดข้อมูล

การลดขนาดของข้อมูลเป็นการย่อยขนาดข้อมูลให้มีขนาดเล็กลงกว่าเดิม โดยทั่วไปการลดขนาดข้อมูลประกอบด้วยการอ่านข้อมูลเข้ามาเป็นลำดับ แล้วทำการแปลงข้อมูลเหล่านี้ให้อยู่ในรูปโค้ดต่างๆ ถ้าหากการลดขนาดข้อมูลมีประสิทธิผลโค้ดที่ได้ต้องมีขนาดสั้นกว่าเดิม โดยโค้ดที่ได้จะเป็นอย่างไรนั้นก็ขึ้นกับโมเดล **(Model)** ที่ใช้ ซึ่งโมเดลคือกลุ่มของข้อมูลที่เลือกไว้และกฎที่ใช้ในการประมวลผลข้อมูลเหล่านั้น เพื่อที่จะหาโค้ดที่สามารถแทนข้อมูลเหล่านั้นได้โปรแกรมจะใช้โมเดลในการคำนวณหาความน่าจะเป็นของการซ้ำซ้อนกันของข้อมูลแต่ละตัว เพื่อทำการสร้างโค้ดสำหรับข้อมูลนั้นๆ ในด้านการลดขนาดของข้อมูลโดยส่วนใหญ่เรามักจะใช้คำว่า **Coding** เพื่อ

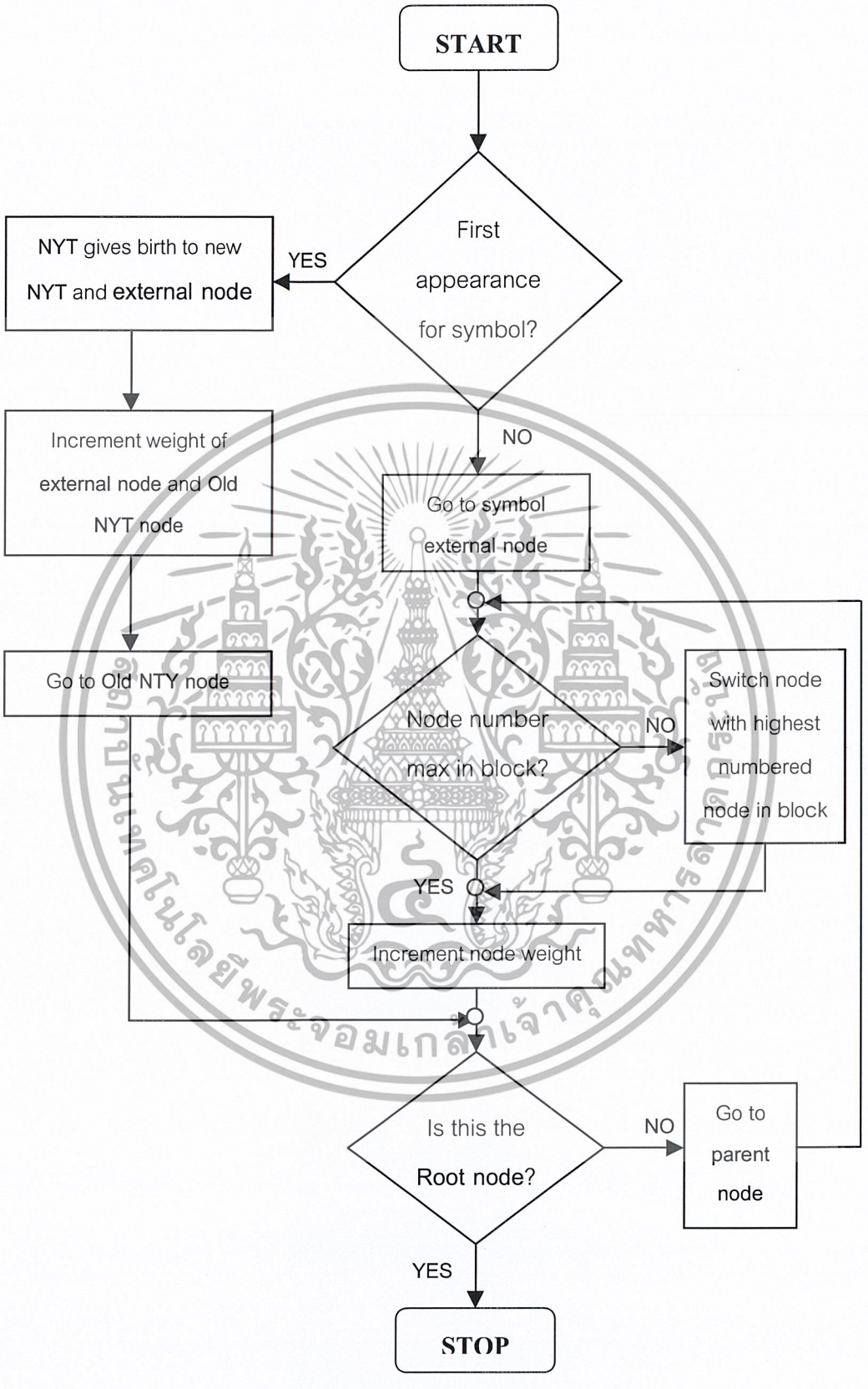
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายถึงกระบวนการลดขนาดของข้อมูลทั้งกระบวนการ แทนที่จะหมายถึงส่วนหนึ่งส่วนใดของกระบวนการ เช่น เราอาจจะได้ยินคำว่า **Huffman Coding** ถูกใช้ในการอธิบายเทคนิคการลดขนาดข้อมูล แต่ในความเป็นจริงจะเป็นเพียงวิธีการเข้ารหัสข้อมูลที่ถูกนำมาใช้ร่วมกับโมเดลในการลดขนาดข้อมูลเท่านั้น ดังนั้นจึงสามารถนิยามการลดขนาดข้อมูลได้ดังนี้

Data Compression = Modeling + Coding **(2.1)**

ถ้าเราดูตัวอย่างจากขั้นตอนวิธีของ **Adaptive Huffman Coding** กระบวนการลดขนาดข้อมูลสามารถแบ่งออกได้เป็นขั้นตอนย่อยๆ แสดงได้ดัง **รูปที่ 2.1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แผนผังการลดขนาดข้อมูลด้วยวิธี Adaptive Huffman Coding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การกำหนดโค้ด **(Coding)**

ในการลดขนาดข้อมูลแบบชนิดตัวอักษรเราต้องการสร้างโค้ดสำหรับแต่ละตัวอักษรโดยใช้จำนวนบิตตามความเป็นจริง เช่น ตัวอักษร c ใช้ 6 บิต หรือตัวอักษร a ใช้ 4 บิต เป็นต้น แต่ในทางปฏิบัติถ้าหากใช้วิธีการเข้ารหัสตัวอักษรโดยใช้รหัสแอสกี **(ASCII)** เราไม่สามารถเข้ารหัสทุกๆ ตัวอักษรได้อย่างกะทัดรัดที่สุดเพราะโดยปกติในรหัสแอสกีแต่ละตัวอักษระใช้ 8 บิต ซึ่งในการแก้ไขปัญหานี้เพื่อให้ได้โค้ดที่สั้นจึงเลือกใช้เทคนิค **Adaptive Huffman Coding** ที่เป็นวิธีการที่พัฒนาขึ้นมาจากเทคนิค **Huffman Coding** เพื่อสร้างโค้ดที่มีความยาวที่แปรเปลี่ยนได้ โดยที่จำนวนบิตของโค้ดแต่ละตัวจะขึ้นอยู่กับความน่าจะเป็นของการซ้ำซ้อนของข้อมูลในกลุ่มนั้น

2.2.2 การทำโมเดล **(Modeling)**

ถ้าหากเปรียบเทียบให้การลดขนาดข้อมูลเป็นรถจักรยานยนต์ **Coding** จะเป็นส่วนวงล้อและ **Modeling** ก็จะเป็นส่วนเครื่องยนต์ ดังนั้นถ้าหากเราเลือกโมเดลที่ไม่ดีพอในการป้อนข้อมูลให้กับส่วนเข้ารหัสอาจจะทำให้การลดขนาดข้อมูลได้ประสิทธิผลต่ำก็ได้ ซึ่งการลดขนาดข้อมูลในประเภท **Lossless Data Compression** โดยทั่วไปจะมีการเลือกใช้เพียง 2 โมเดลคือ **Statistical** และ **Dictionary-Based** โดยที่โมเดล **Statistical** จะอ่านข้อมูลและทำการเข้ารหัสแต่ละตัวอักษรในเวลาเดียวกัน ซึ่งโค้ดที่ได้จะขึ้นกับความน่าจะเป็นของการซ้ำซ้อนของแต่ละตัวอักษร ตัวอย่างของวิธีที่ใช้โมเดลชนิดนี้ก็คือ **Huffman Coding** และ **Adaptive Huffman Coding** ส่วนโมเดล **Dictionary-Based** จะใช้โค้ดตัวหนึ่งที่มีในศัพท์ **(Dictionary)** ในการแทนที่สตริงชุดหนึ่งๆ โดยถ้าหากพบสตริงดังกล่าวผลลัพธ์ที่ได้ก็จะเป็นค่าอินเด็กซ์ **(index)** หรือค่าพอยเตอร์ **(pointer)** สำหรับตำแหน่งที่พบในคำศัพท์แทนที่จะเป็นโค้ดสำหรับสตริงนั้นๆ

2.3 ทฤษฎีแผนภูมิต้นไม้ (Tree)

นิยาม

แผนภูมิต้นไม้ (Tree) คือ เซตจำกัดของโหนดที่มีสมาชิกตั้งแต่ 1 ขึ้นไป และต้องมีด้านเชื่อมระหว่างโหนดซึ่งไม่มีลูป **(loop)** หรือ **circuit** และมีโหนดพิเศษที่เรียกว่าโหนดราก**(Root)**

โหนดราก (Root) คือ โหนดบนสุดของแผนภูมิต้นไม้

Parent ของโหนดใดๆ คือ โหนดที่มีด้านเชื่อมซึ่งมีทิศทางจากโหนดนั้นต่อไปยังโหนดอื่นๆ

Children ของโหนดใดๆ คือ โหนดที่มีด้านเชื่อมต่อจาก **Parent node**

ดีกรีของโหนด (Degree) คือ จำนวน **child** ของแต่ละโหนด

โหนดใบ (Leaf node) คือ โหนดที่มีดีกรีเป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Internal node คือ โหนดที่มีดีกรีตั้งแต่ 1 ขึ้นไป

Sibling คือ Children ที่มี **Parent** เดียวกัน

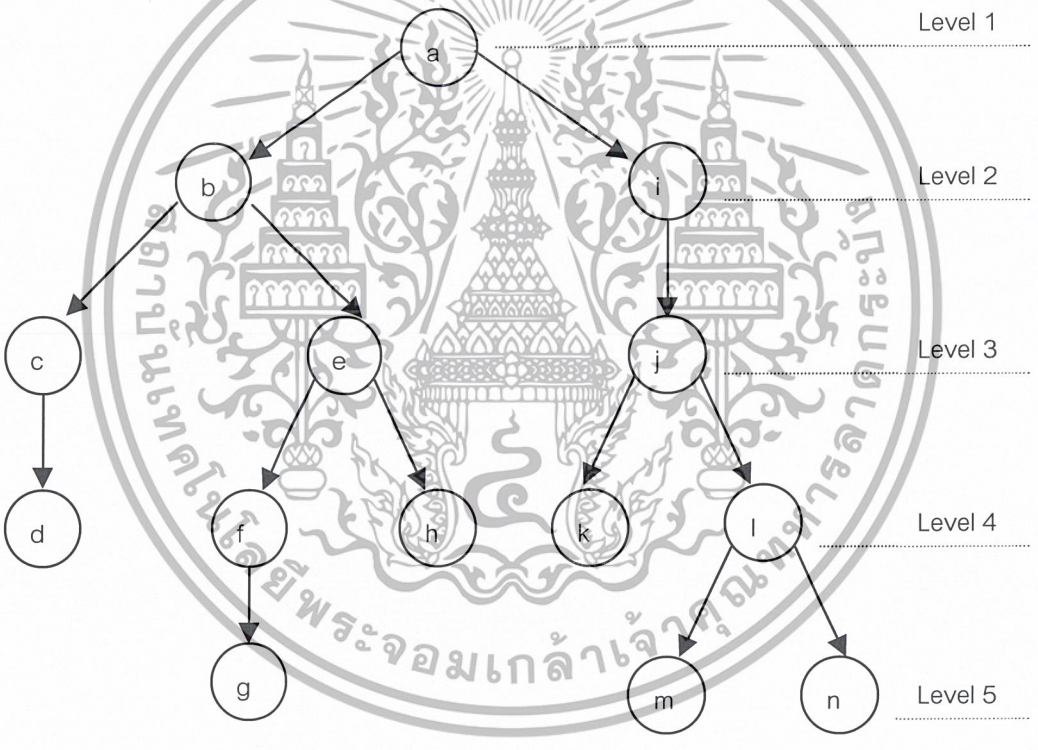
ดีกรีของแผนภูมิต้นไม้ (Degree of Tree) คือดีกรีที่มากที่สุดของแผนภูมิต้นไม้นั้นๆ

Ancestor ของโหนดใดๆ คือ ทุกๆ โหนดที่อยู่ในเส้นทางเดียวกันที่แยกตัวออกจากโหนดรากจนถึงโหนดนั้นๆ

Descendent ของโหนดใดๆ คือ ทุกๆ โหนดที่มีโหนดนั้นๆ เป็น **ancestor**

ระดับ (Level) จะเริ่มนับตั้งแต่โหนดรากโดยมีระดับ **(Level)** เท่ากับ 1 (ดูตัวอย่างจาก **รูปที่ 2.2**)

ความสูง **(Height)** คือ ระดับสูงสุดของโหนดใดๆ ในแผนภูมิต้นไม้

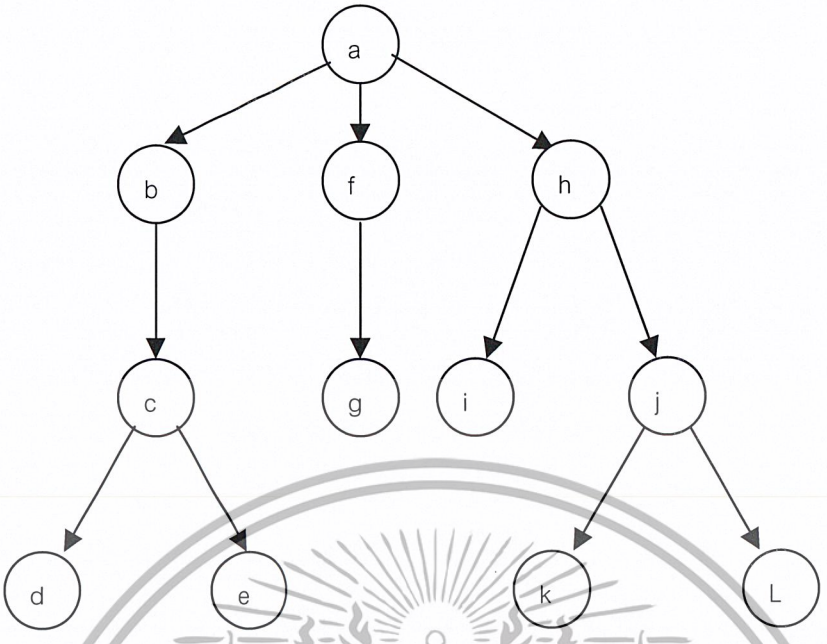


รูปที่ 2.2 ตัวอย่างแผนภูมิต้นไม้

จาก**รูปที่ 2.2** จะได้ **Level** เท่ากับ 5 ดังนั้นแผนภูมิต้นไม้นี้มีความสูง **(height)** เท่ากับ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.1



รูปที่ 2.3 แผนภูมิต้นไม้ในตัวอย่างที่ 2.1

จากตัวอย่างแผนภูมิไม้รูปที่ 2.3 ซึ่งมีโหนด a เป็น **root** สามารถหา **parent** ของโหนด g, **children** ของโหนด h, **sibling** ของโหนด I, **ancestors** ทั้งหมดของโหนด d, **descendants** ทั้งหมดของโหนด b, **internal vertices** ทั้งหมดของแผนภูมิต้นไม้ และ **leaves** ทั้งหมดของแผนภูมิต้นไม้ได้ดังต่อไปนี้

parent ของโหนด g คือโหนด f

children ของโหนด h คือโหนด I, j

sibling ของโหนด I คือโหนด j

ancestors ทั้งหมดของโหนด d คือโหนด a, b, c

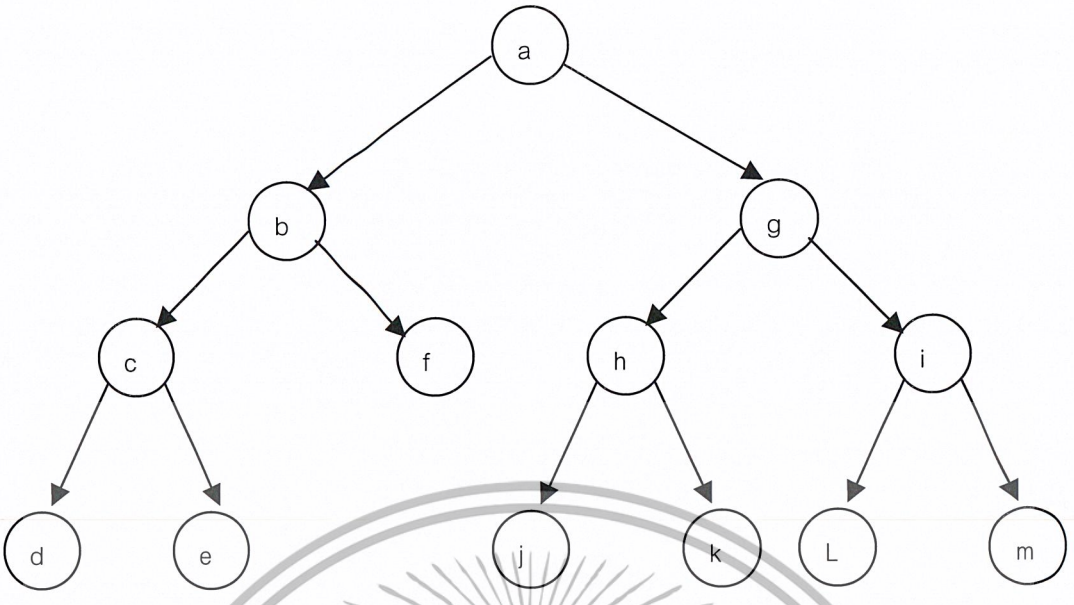
descendants ทั้งหมดของโหนด b คือโหนด c, d, e

internal vertices ทั้งหมดคือโหนด a, b, c, f, h, j

leaves ทั้งหมดคือโหนด d, e, g, I k, L

สำหรับในปัญหาพิเศษฉบับนี้ใช้แผนภูมิต้นไม้แบบ **Binary tree** ซึ่งเป็นแผนภูมิต้นไม้ที่โหนดทุกโหนดที่เป็น **internal node** มี **children** เท่ากับ 2 โดย **child** ของแต่ละ **internal node** มีอันดับจากซ้ายไปขวาเรียก **first child** ว่า **left child** เรียก **second child** ว่า **right child** จาก**รูปที่ 2.4** จะได้ว่าโหนด d เป็น **left child** และ โหนด e เป็น **right child** ของโหนด c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ตัวอย่างแผนภูมิต้นไม้แบบ Binary tree

2.4 ทฤษฎีความน่าจะเป็นในการลดขนาดข้อมูล

ขั้นตอนการลดขนาดของข้อมูลโดยทั่วไปแล้วมีอยู่หลายขั้นตอน โดยจะมีขั้นตอนหลักๆ คือการหาค่าความถี่และค่าความน่าจะเป็นของข้อมูล โดยที่ข้อมูลที่มีการใช้มากจะมีการลดขนาดให้มีขนาดของบิตต่ำในแต่ละตัวอักษรส่วนข้อมูลที่มีการใช้น้อยให้มีขนาดบิตสูง หากต้องการที่จะรู้ว่าข้อมูลตรงส่วนใดที่มีการใช้มากหรือใช้น้อย ก็จะใช้ทฤษฎีความน่าจะเป็นทำการเช็คข้อมูลโดยการเก็บค่าความถี่และความน่าจะเป็นของข้อมูลแต่ละตัว ดังนั้นวิธีการลดขนาดข้อมูลโดยใช้วิธี **Adaptive Huffman** ก็เช่นเดียวกันจะใช้วิธีเก็บความถี่และคำนวณค่าความน่าจะเป็น โดยความน่าจะเป็นของเหตุการณ์ใดๆ มีค่าได้ตั้งแต่ 0 ถึง 1 และค่าความน่าจะเป็นคือตัวเลขซึ่งชี้บอกถึงเหตุการณ์นั้นๆ ว่ามีโอกาสเกิดขึ้นมากน้อยเพียงใด ถ้ามีค่าใกล้ 1 หมายถึงเหตุการณ์นั้นจะมีโอกาสในการเกิดสูงและจะเกิดขึ้นแน่นอนถ้ามีค่าเป็น 1 (หรือมีโอกาสเกิด 100%) ถ้าความน่าจะเป็นมีค่าใกล้ 0 แสดงว่าเหตุการณ์นั้นมีโอกาสเกิดขึ้นน้อยมาก และจะไม่เกิดขึ้นหรือเป็นไปไม่ได้ถ้ามีค่าเป็น 0 ความน่าจะเป็นที่ใช้ในการลดขนาดข้อมูลจะคำนวณได้ดังสมการต่อไปนี้

$$\text{ความน่าจะเป็นของข้อมูลแต่ละตัวอักษร} = \frac{\text{(จำนวนครั้งที่ใช้ข้อมูลแต่ละตัว)}}{\text{(จำนวนครั้งที่ใช้ข้อมูลทั้งหมด)}} \qquad (2.16)$$

ตัวอย่างที่ 2.2

มีข้อมูลชุดหนึ่งซึ่งมีตัวอักษรดังนี้ “ACDCBAEACB” ความน่าจะเป็นของข้อมูลแต่ละตัวอักษรของข้อมูลนี้ ซึ่งมีจำนวนตัวอักษรทั้งหมด 10 ตัว เป็นตัวอักษร A 3 ตัวอักษร เป็นตัวอักษร B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ตัว เป็นตัวอักษร C 3 ตัว เป็นตัวอักษร D 1 ตัวอักษร และเป็นตัวอักษร E 1 ตัวอักษร จะได้ความน่าจะเป็นดังตารางที่ 2.1

ตารางที่ 2.1 ตารางตัวอย่างของการหาค่าความน่าจะเป็นจากตัวอย่างที่ 2.2

Alphabet	Probability of alphabet
A	3/10
B	2/10
C	3/10
D	1/10
E	1/10

2.5 การลดขนาดข้อมูลโดยแนวคิดทางคณิตศาสตร์

ทางด้านทฤษฎีของข้อมูลแนวคิดพื้นฐานในการลดขนาดข้อมูลก็คือ "ถ้าทราบความน่าจะเป็นของสัญลักษณ์หรือตัวอักษรในข้อมูลชุดหนึ่ง ก็จะสามารถลดขนาดข้อมูลชุดนั้นได้" ดังนั้นแนวคิดในการสร้างขั้นตอนวิธีการ **(Algorithms)** ในการลดขนาดของข้อมูลหลายชนิดในปัจจุบันถูกพัฒนามาซึ่งจะยกตัวอย่างมา 2 วิธีที่ใช้หลักการทางคณิตศาสตร์เกี่ยวกับความน่าจะเป็น

2.5.1 วิธี Huffman Coding

วิธีการเข้ารหัสข้อมูลที่มีประสิทธิภาพวิธีหนึ่งซึ่งเป็นที่รู้จักกันดีก็คือ **Huffman Coding** ซึ่งได้ถูกคิดค้นโดย **D.A. Huffman** และได้ถูกตีพิมพ์ครั้งแรกในบทความ **A Method for the Construction of Minimum Redundancy Codes** ในปี ค.ศ. **1952** ซึ่งวิธีการนี้สามารถสร้างโค้ดที่มีความยาว(จำนวนบิต) แตกต่างกันได้ สัญลักษณ์หรือตัวอักษรที่มีความน่าจะเป็นสูงกว่าก็จะถูกเข้ารหัสด้วยโค้ดที่มีจำนวนบิตน้อยกว่าและที่สำคัญโค้ดทุกตัวที่ได้จะมีแอตทริบิวต์ทำหน้าที่ไม่เหมือนกันดังนั้นโค้ดที่ได้จึงสามารถถูกถอดรหัสได้อย่างถูกต้องส่วนขั้นตอนการถอดรหัสโค้ดโดยวิธี **Huffman Coding** โดยส่วนใหญ่มักกระทำโดยใช้แผนภูมิต้นไม้ **(Binary Tree)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนวิธีการของ **Huffman Coding (Huffman Algorithm)**

1. สำหรับข้อมูลชุดหนึ่งนับจำนวนความถี่หรือความน่าจะเป็นของตัวอักษรแต่ละตัว

2. เรียงลำดับ **(sort)** ตามค่าความน่าจะเป็นจากน้อยไปหามากเก็บไว้ในลิสต์ **(list)**

3. โหนดจะถูกสร้างทีละคู่ โดยเริ่มจากคู่ของข้อมูลที่มีค่าความน่าจะเป็นน้อยที่สุดไปหาคู่ของข้อมูลที่มีค่าความน่าจะเป็นหรือค่าน้ำหนักมากที่สุด แต่ถ้าหากข้อมูลตัวสุดท้ายไม่สามารถจัดเข้าคู่ได้คือเหลือตัวเดียว ก็จะถูกแยกเป็นโหนดอิสระที่มีกิ่งเพียงกิ่งเดียว

4. สร้างโหนดแม่ **(Parent Node)** จากโหนดลูกคู่ที่ได้จากขั้นตอนที่ 3 ซึ่งจะมีค่าน้ำหนักเท่ากับผลรวมของค่าน้ำหนักของโหนดลูก **(Child Node)** ทั้งสอง

5. โหนดแม่ที่ได้ก็จะถูกเพิ่มเข้ามาเป็นโหนดอิสระในลิสต์ที่เรียงจากน้อยไปมาก และโหนดลูกก็จะถูกยกเลิกออกจากลิสต์

6. การกำหนดค่าบิตของโหนดลูกคู่ที่รวมกันเป็นโหนดแม่ในขั้นตอนที่ 4 กำหนดโดยโหนดลูกจะถูกกำหนดให้มีค่าเป็น 0 (ด้านซ้าย) ส่วนโหนดลูกที่เหลือ (ด้านขวา) ก็จะถูกกำหนดให้มีค่าเป็น 1

7. ให้ทำซ้ำขั้นตอนที่ 3 ถึง 6 จนกระทั่งเหลือโหนดอิสระเพียงโหนดเดียวซึ่งจะถูกกำหนดให้เป็นโหนดราก **(root node)**

ตัวอย่างที่ 2.3 การสร้างแผนภูมิต้นไม้โดยวิธี Huffman

สมมติว่าข้อมูลชุดหนึ่งประกอบด้วยตัวอักษร A ถึง H เป็นจำนวน 53 ตัวอักษร มีลักษณะดังนี้

DAABCGCBEABBADACEBBEDECFCACADAFBADEGDCAAACADBAAACAACH

ซึ่งหลังจากนับความถี่และเรียงลำดับจากมากไปน้อยในขั้นตอนที่ 1 และ 2แล้วได้ค่าความถี่ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 แสดงจำนวนความถี่ของตัวอักษรในข้อมูลชุดหนึ่งจากตัวอย่างที่ 2.3

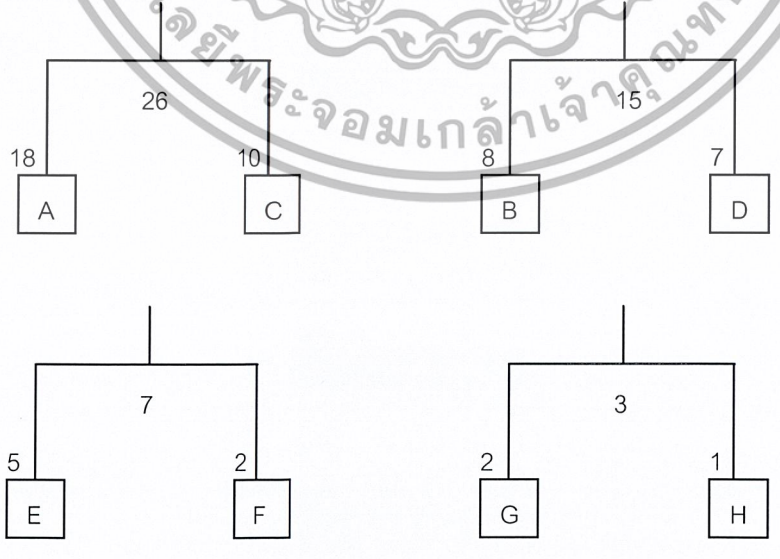
ตัวอักษร	ความถี่
A	18
C	10
B	8
D	7
E	5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงจำนวนความถี่ของตัวอักษรในข้อมูลชุดหนึ่งจากตัวอย่างที่ 2.3 (ต่อ)

ตัวอักษร	ความถี่
F	**2**
G	**2**
H	**1**

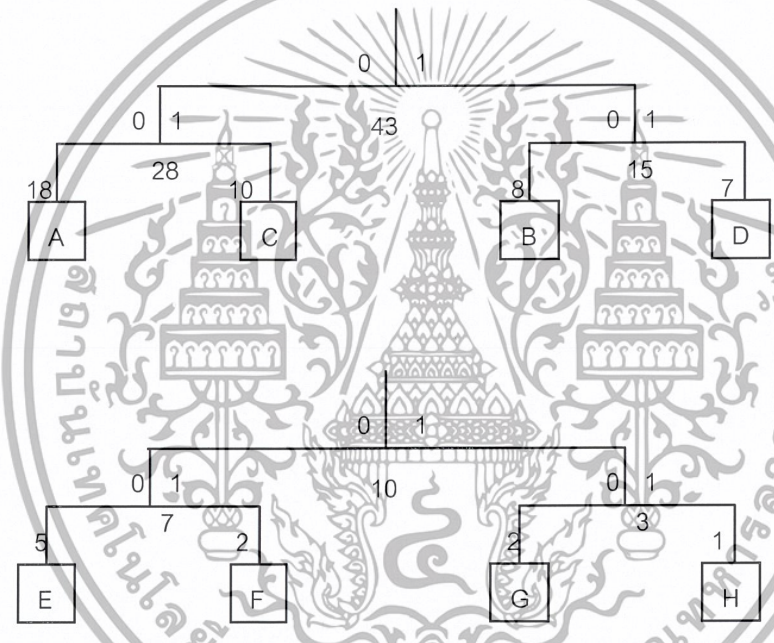
ในขั้นตอนแรก (ขั้นตอนที่ 3 ใน **Huffman Coding**) จะมีโหนดลูกทั้งหมด 8 โหนดในลิสต์ (H,G,F,E,D,B,C,A) และเนื่องจากจำนวนข้อมูลเป็นเลขคู่ดังนั้นจึงสามารถจับคู่เพื่อสร้างโหนดคู่ได้ 4 คู่โดยการจับคู่โหนดจะต้องเริ่มจากคู่ที่มีค่าผลรวมของค่าความน่าจะเป็นน้อยที่สุดไปยังค่ามากที่สุด ดังนั้นโหนดคู่แรกจึงสามารถจับคู่ได้ระหว่าง G กับ H (หรือ F กับ H เพราะ G กับ F มีค่าความถี่เท่ากัน) ซึ่งมีค่าน้ำหนักเท่ากับผลรวมระหว่าง 2 กับ 1 เป็น 3 โหนดคู่ต่อมาก็จะเป็นการจับคู่ระหว่าง E กับ F ซึ่งมีค่าน้ำหนักเท่ากับผลรวมระหว่าง 5 กับ 2 เป็น 7 โหนดคู่ต่อมาก็จะเป็นการจับคู่ระหว่าง B กับ D ซึ่งมีค่าน้ำหนักเท่ากับผลรวมระหว่าง 8 กับ 7 เป็น 15 และโหนดคู่สุดท้ายก็จะเป็นการจับคู่ระหว่าง A กับ C ซึ่งมีค่าน้ำหนักเท่ากับผลรวมระหว่าง 18 กับ 10 เป็น 28 จากนั้นก็ทำการสร้างโหนดแม่โดยเกิดจากผลรวมของโหนดลูกคู่ต่างๆ ดังนั้นโหนดแม่ของโหนดคู่ G กับ H ก็จะมีค่าน้ำหนักเป็น 3 โหนดแม่ของโหนดคู่ E กับ F ก็จะมีค่าน้ำหนักเป็น 7 โหนดแม่ของโหนดคู่ B กับ D ก็จะมีค่าน้ำหนักเป็น 15 และโหนดแม่ของโหนดคู่ A กับ C ก็จะมีค่าน้ำหนักเป็น 28 และต่อจากนั้นทำการกำหนดค่าบิตให้กับแต่ละกิ่ง ดังนั้นกิ่ง A,B,E,G จึงมีค่าบิต 0 และกิ่ง C,D,F,H ก็จะมีค่าบิตเป็น 1 ซึ่งแผนภูมิต้นไม้ที่ได้จากรอบแรกก็จะมีลักษณะดัง **รูปที่ 2.5**



รูปที่ 2.5 แผนภูมิต้นไม้ Huffman หลังจากรอบแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

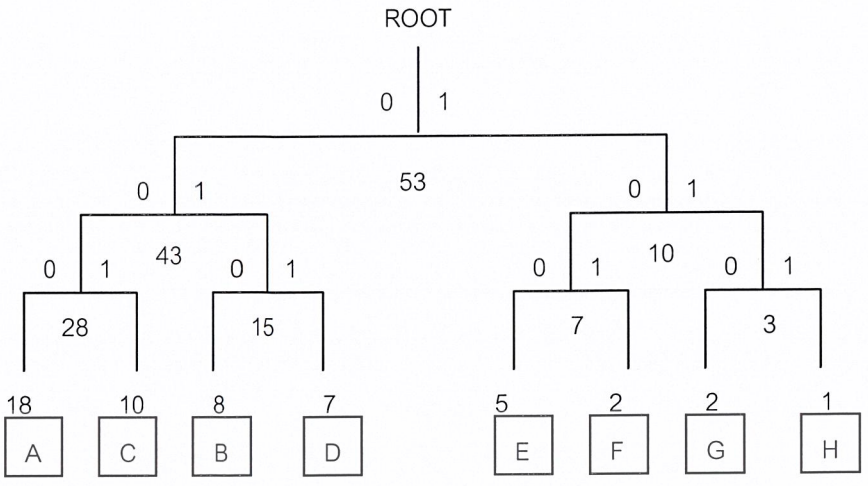
ต่อไปเพิ่มโหนดแม่ 4 โหนด (G กับ H, E กับ F, B กับ D และ A กับ C) เข้าไปในลิสต์และยกเลิกโหนดลูก (A,B,C,D,E,F,G,H) ออกจาลิสต์ ดังนั้นค่าในลิสต์คือ G กับ H, E กับ F, B กับ D และ A กับ C 3, 7, 15 และ28 ตามลำดับรวมโหนดคู่จากโหนดแม่ในลิสต์ที่เรียงจากน้อยไปมาก โดยสมมติว่าโหนดแม่ที่ได้ก็คือโหนดลูกของลำดับถัดไปนั่นเอง ดังนั้นโหนดแม่ของโหนดคู่ E กับ F และ G กับ H ก็จะถูกสร้างขึ้นมาโดยมีค่าน้ำหนักเท่ากับ 7 + 3 เท่ากับ 10 และโหนดแม่ของโหนดคู่ A กับ C และ B กับ D ก็จะถูกสร้างขึ้นมาโดยมีค่าน้ำหนักเท่ากับ 28 + 15 เท่ากับ 43 ดังนั้นกิ่งโหนดแม่ A กับ C และ E กับ F ก็จะถูกกำหนดให้มีค่าบิตเท่ากับ 0 และกิ่งของโหนดแม่ B กับ D และ G กับ H ก็จะถูกกำหนดให้มีค่าเท่ากับ 1 ซึ่งแผนภูมิต้นไม้ที่ได้จากรอบที่สองจะมีลักษณะดัง **รูปที่ 2.6**



รูปที่ 2.6 แผนภูมิต้นไม้ของวิธี Huffman หลังจากรอบที่สอง

ต่อไปเพิ่มโหนดแม่ 2 โหนด (AC กับ BD) และ (EF กับ GH) เข้าไปในลิสต์และยกเลิกโหนดลูกออกจากลิสต์ ดังนั้นค่าในลิสต์คือ AC กับ BD และ EF กับ GH ซึ่งมีค่าน้ำหนัก 43 และ 10 ตามลำดับนำมาสร้างโหนดคู่ในขั้นตอนสุดท้ายของการสร้างแผนภูมิต้นไม้เพื่อให้เหลือโหนดรากเพียงโหนดเดียว ดังนั้นโหนดแม่ AC กับ BDนำมาจับคู่กับโหนดแม่ EF กับ GH ซึ่งโหนดรากที่ได้ก็จะมีค่าน้ำหนักเป็น 43 + 10 เท่ากับ 53 และกิ่งที่ได้จากโหนดแม่ของ AC กับ BC ก็จะถูกกำหนดให้มีค่าบิตเป็น 0 ส่วนกิ่งที่ได้จากโหนดแม่ของ EF กับ GH ก็จะถูกกำหนดให้มีค่าบิตเป็น 1 ซึ่งแผนภูมิต้นไม้ที่ได้จากขั้นตอนที่สามก็จะมีลักษณะดัง**รูปที่ 2.17**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แผนภูมิต้นไม้ Huffman

จากแผนภูมิข้างต้น เราสามารถที่จะเข้ารหัสหรือถอดรหัสสำหรับตัวอักษรแต่ละตัวได้โดยง่ายโดยการเดินตามกิ่งสาขาของต้นไม้ซึ่งจะได้โค้ดดัง**ตารางที่ 2.3**

ตารางที่ 2.3 แสดงโค้ดของวิธี Huffman

ตัวอักษร	A	B	C	D	E	F	G	H
โค้ด	000	010	001	011	100	101	110	111

จะเห็นได้ว่าข้อมูลที่มีค่าความถี่มากกว่าจะได้โค้ดมีจำนวนบิตน้อยกว่า และเมื่อเปรียบเทียบกับการใช้อักษรรหัสแอสกี (ASCII) ขนาด 8 บิต ซึ่งข้อมูลทั้งหมดจะมีความยาวเท่ากับ 8 x 53 เท่ากับ 424 บิต จะพบว่าโค้ดที่ได้จากวิธี **Huffman** จะทำให้ข้อมูลทั้งหมดมีความยาวเพียง 54 + 30 + 24 + 21 + 15+ 6 + 6 +3 = 159 บิตเท่านั้น

2.5.2 วิธี **Adaptive Huffman Coding**

วิธี **Adaptive Huffman Coding** ต่างจากวิธี **Huffman Coding** ในเวลาเข้ารหัส โดยวิธีแบบ **Huffman Coding** นั้นจะนำข้อมูลทั้งหมดมาวิเคราะห์แล้วจึงค่อยสร้างแผนภูมิต้นไม้ โดยใช้วิธีสร้างจากล่างขึ้นบน **(Bottom-Up)** แต่วิธี **Adaptive Huffman Coding** นั้น จะทำการอ่านข้อมูลมาทีละตัวแล้วนำมาสร้างแผนภูมิต้นไม้ทันที โดยจะใช้วิธีสร้างแบบบนลงล่าง **(Top-Down)** และจะมีการเปลี่ยนแปลง **(Update)** แผนภูมิต้นไม้ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนวิธีของ **Adaptive Huffman Coding (Adaptive Huffman Algorithm)**

1. อ่านข้อมูลขึ้นมาทีละตัวอักษรนำมาตรวจสอบว่ามีอยู่ใน **Tree** ก่อนแล้วหรือไม่ โดยเริ่มอ่านตั้ง แต่ตัวแรก

2. ถ้ายังไม่เคยมีอยู่ใน **Tree** ให้ไปที่ โหนดว่าง **(NYT Node)** และสร้างโหนดลูก **(Child Node)** ขึ้นมาใหม่ โดยให้โหนดลูกด้านซ้ายเป็นโหนด **NYT** ตัวใหม่และให้ค่าน้ำหนัก **(Weight)** เป็น 0 ส่วนโหนดลูกด้านขวาให้เป็นโหนดที่เก็บค่าตัวอักษรตัวนั้นและจะมีค่าน้ำหนักเป็น 1 เสร็จแล้วให้ข้ามไปทำข้อ 4

3. ถ้ามีอยู่ก่อนแล้ว ให้ไปที่โหนด **(Node)** ของตัวอักษรนั้นแล้วเพิ่มค่าน้ำหนักขึ้นอีก 1

4. นำค่าน้ำหนักของของโหนดตัวอักษรนั้นไปเทียบกับค่าน้ำหนักของโหนดอื่นจนกว่าจะพบโหนดที่มีค่าน้ำหนักมากกว่าตัวแรกหรือจนกว่าจะพบโหนดราก **(Root Node)** โดยไล่จากโหนดที่อยู่ทางขวามือของโหนดตัวอักษรนั้นไปทางขวาเรื่อย ๆ ถ้าหาจนถึงตัวขวาสุดของชั้น **(Level)** นั้นแล้วยังไม่พบให้ไปหาต่อที่ชั้นถัดไปด้านบนโดยไล่จากซ้ายไปขวา

5. ถ้าโหนดด้านซ้ายมือของโหนดที่มีค่าน้ำหนักมากกว่า มีค่าน้ำหนักน้อยกว่าค่าน้ำหนักของโหนดตัวอักษรตัวนั้นให้ทำการสลับที่โหนดตัวอักษรกับโหนดนั้น (โดยย้ายโหนดที่เป็น **Descendent** ตามไปด้วย) แต่ถ้ามีค่าเท่ากันไม่ต้องทำอะไร

6. เพิ่มค่าน้ำหนักให้กับโหนดแม่ **(Parent Node)** ของโหนดตัวอักษรนั้นขึ้นอีก 1 แล้วย้อนกลับไปทำตั้งแต่ข้อ 4 ใหม่ โดยใช้ค่าน้ำหนักของโหนดแม่ตัวนั้น ทำอย่างนี้ไปเรื่อย ๆ จนกว่าจะพบโหนดราก

7. อ่านตัวอักษรตัวถัดไปแล้วกลับไปเริ่มทำตั้งแต่ข้อ 1 ใหม่จนกว่าข้อมูลจะหมด

ตัวอย่างที่ 2.4 การสร้างแผนภูมิต้นไม้ด้วยวิธี Adaptive Huffman Coding

สมมุติว่ามีข้อมูลอยู่ชุดหนึ่งประกอบด้วยตัวอักษร A ถึง H เป็นจำนวนทั้งหมด 53 ตัวอักษร มีลักษณะดังนี้

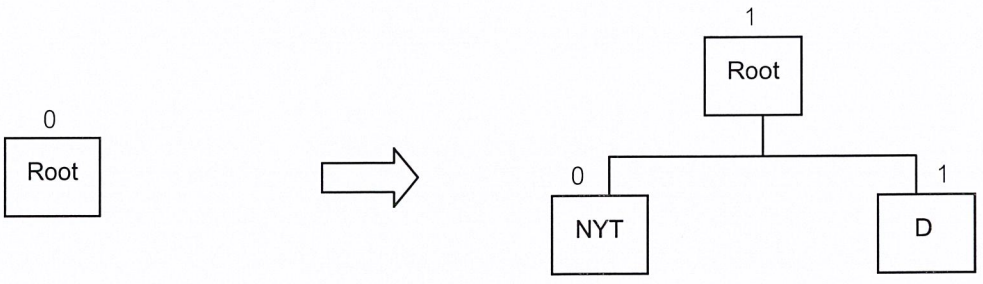
DAABCGCBEABBADACEBBEDECFCACADAFBADEGDCAAACADBAAACAACH

ขั้นตอนการสร้างแผนภูมิต้นไม้มีดังนี้

1. อ่านตัวอักษรขึ้นมาทีละตัว โดยเริ่มจากตัวแรกสุดได้แก่ตัวอักษร D เมื่อนำไปตรวจสอบดูปรากฏว่าไม่เคยพบมาก่อนเพราะเป็นตัวอักษรตัวแรก ดังนั้นให้สร้างโหนดลูกต่อออกมาจากโหนดราก โดยให้โหนดด้านซ้ายเป็นโหนดว่างซึ่งจะมีค่าน้ำหนักเป็น 0 ส่วนโหนดด้านขวาให้เป็นโหนดของตัวอักษรนั้น

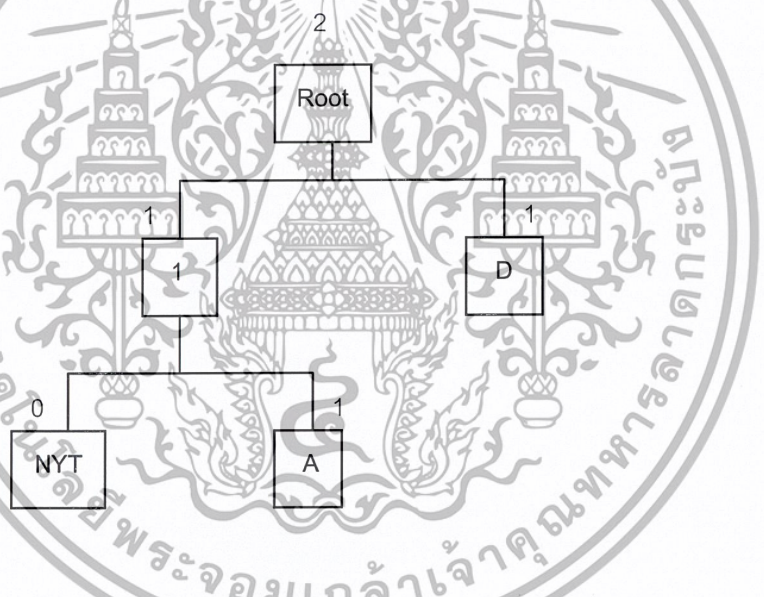
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.8 แผนภาพแสดงการเปลี่ยนแปลงของ Tree หลังจากพบตัวอักษรตัว D

2. อ่านตัวอักษรตัวถัดไป ได้แก่ A ตรวจแล้วพบว่าไม่มีอยู่ใน **Tree** ก็ต้องทำแบบ **Algorithm** ในข้อ 2 จะได้ดังรูปที่ 2.9

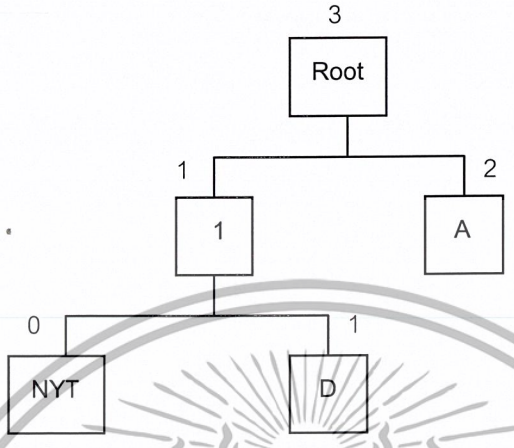


รูปที่ 2.9 แผนภาพแสดง Tree หลังจากพบ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

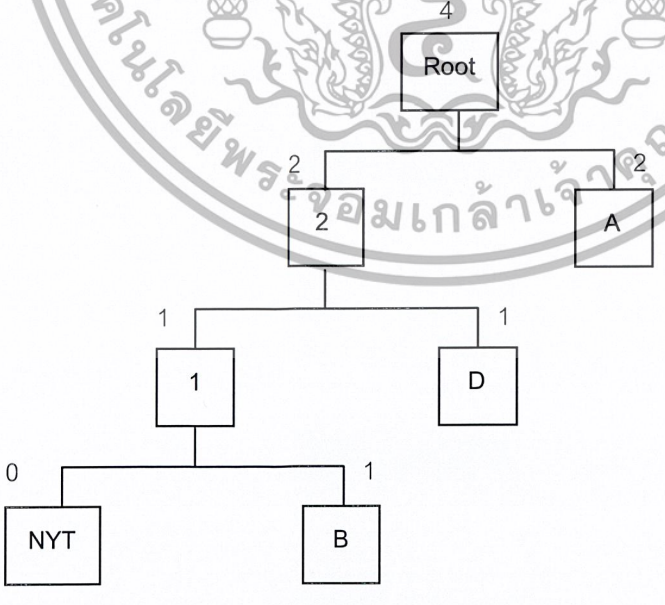
47353

3. อ่านตัวอักษรตัวถัดไปเป็น A ซึ่งเคยมีอยู่ใน **Tree** แล้วให้ทำตาม **Algorithm ข้อ 3** และนำค่าน้ำหนักของ A ไปเปรียบเทียบจะพบว่า A ต้องสลับที่กับ D จะได้แผนภูมิต้นไม้เป็นดังนี้



รูปที่ 2.10 แผนภาพแสดงแผนภูมิต้นไม้หลังจากทำการสลับที่โหนด

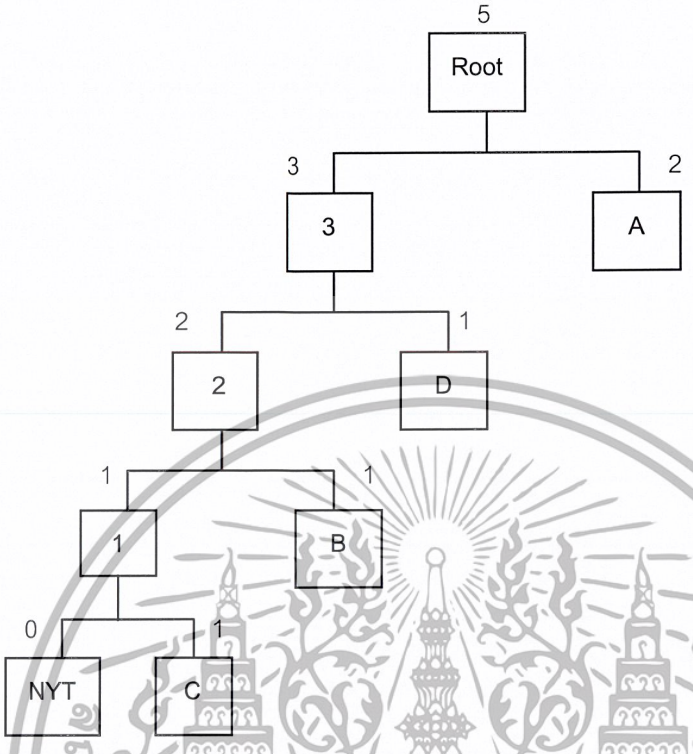
4. อ่านตัวอักษรตัวถัดมาได้แก่ B ซึ่งยังไม่มีใน **Tree** มาก่อน ก็จะทำการสร้างโหนดใหม่ขึ้นมาต่อจากโหนดว่าง จะได้แผนภูมิต้นไม้เป็นเช่นนี้



รูปที่ 2.11 แผนภาพแสดงแผนภูมิต้นไม้เมื่อเพิ่ม B เข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

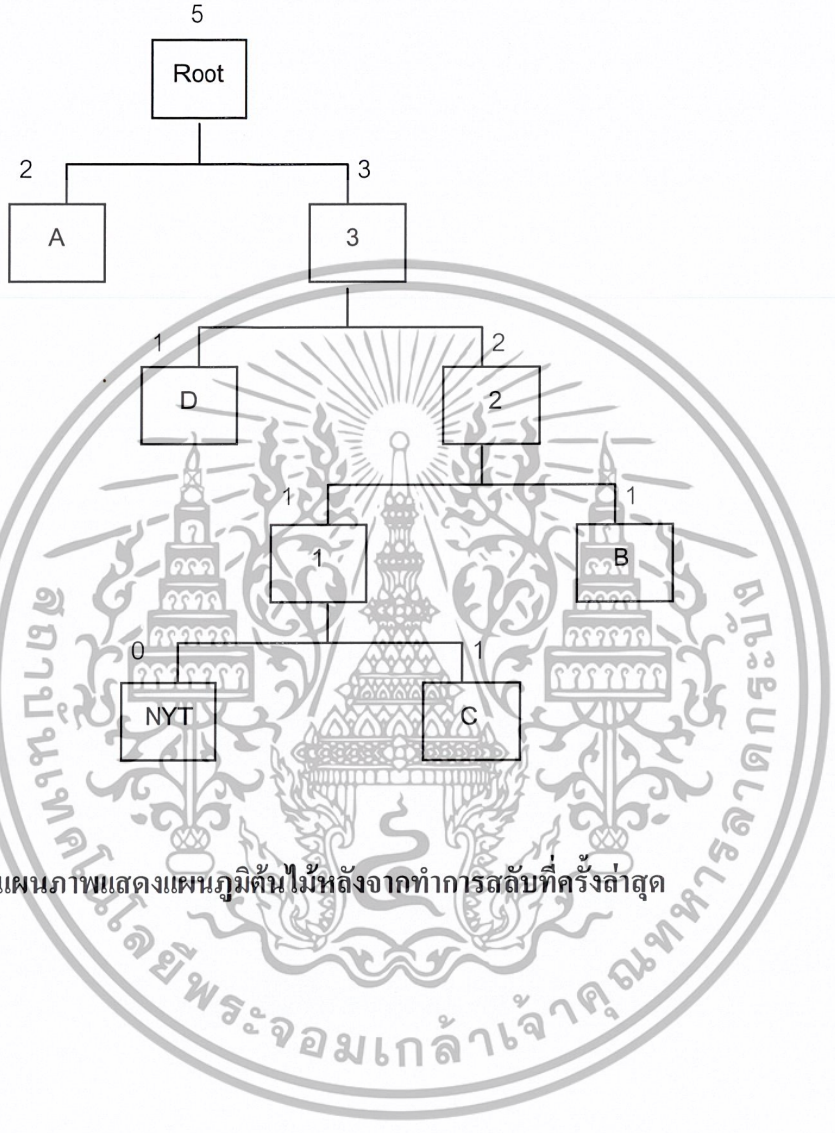
5. อ่านตัวอักษรตัวถัดไป ได้แก่ C ยังไม่มีอยู่ใน Tree ให้ทำเหมือนขั้นตอนเดิม



รูปที่ 2.12 แผนภาพแสดงแผนภูมิต้นไม้หลังจากเพิ่ม C ลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

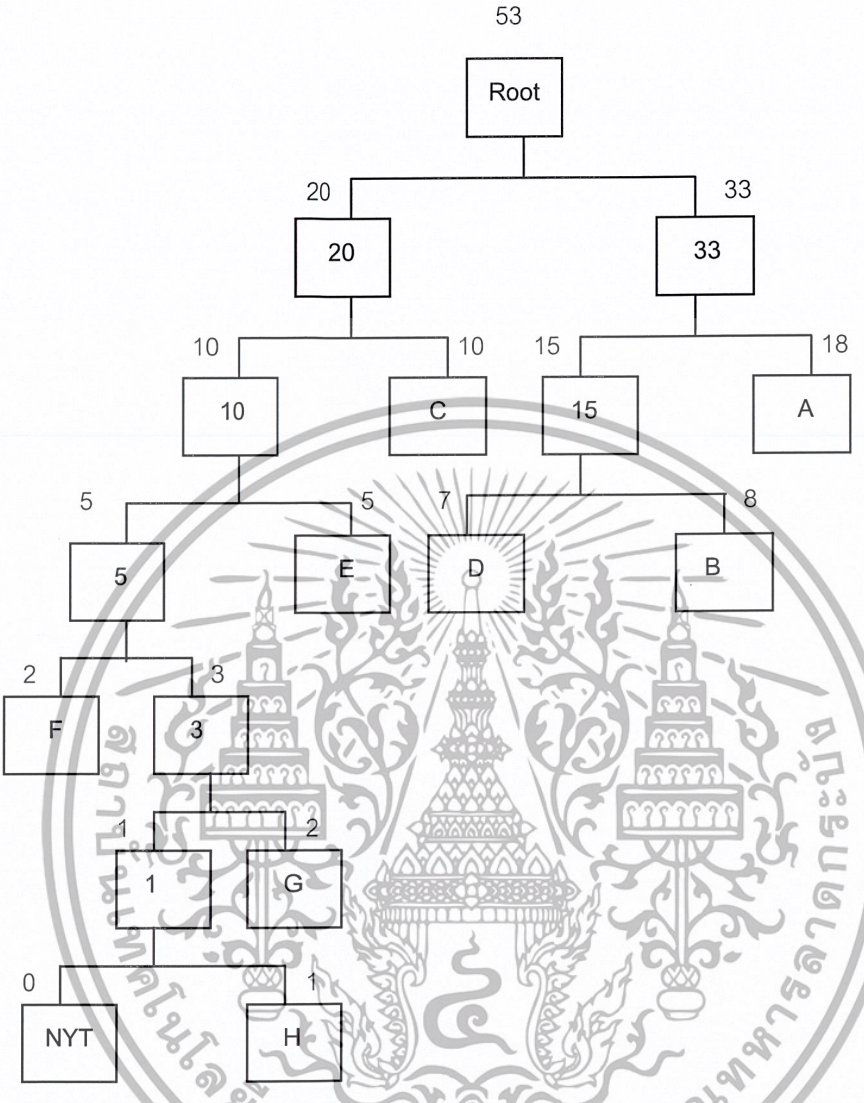
จะเห็นได้ว่าโหนดแม่ของB (2)มีค่าน้ำหนักมากกว่าโหนด D และโหนดแม่ของ D (3) มีค่าน้ำหนักมากกว่าโหนด A จึงทำให้เกิดการสลับที่ 2 ครั้งและแผนภูมิต้นไม้ที่ทำการสลับที่แล้วเป็นดัง**รูปที่ 2.13**



รูปที่ 2.13 แผนภาพแสดงแผนภูมิต้นไม้หลังจากทำการสลับที่ครั้งล่าสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. อ่านข้อมูลไปเรื่อย ๆ จนครบทุกตัวจะได้แผนภาพต้นไม้เป็นดังนี้



รูปที่ 2.14 แผนภาพแสดงแผนภูมิต้นไม้เมื่ออ่านข้อมูลครบทุกตัวแล้ว

จากแผนภูมิต้นไม้ที่ได้ เราจะสามารถเข้ารหัสหรือถอดรหัสสำหรับตัวอักษรแต่ละตัวในข้อมูลชุดนี้ได้ง่ายโดยการเดินตามกิ่งสาขาของต้นไม้ ถ้าเดินไปทางซ้ายจะได้เป็นบิต 0 แต่ถ้าเดินไปทางขวาจะเป็นบิต 1 ซึ่งจะได้โค้ดตาม**ตารางที่ 2.4** ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่2.4 แสดงโค้ดที่แปลงได้จากวิธี Adaptive Huffman Coding

ตัวอักษร	A	B	C	D	E	F	G	H
โค้ด	11	101	10	100	001	0000	00011	000101

จะเห็นได้ว่าข้อมูลหลังจากการแปลงโค้ดแล้วจะมีขนาดเล็กลงกว่าเดิม โดยข้อมูลเดิมจะเก็บเป็นรหัสแอสกี (ASCII) ขนาด 8 บิต ข้อมูลทั้งหมดก็จะมีขนาด 8 x 53 เท่ากับ 424 บิต แต่ข้อมูลที่ได้หลังจากการแปลงแล้วจะมีขนาด 140 บิตเท่านั้น ต่างจากแบบวิธี **Huffman Coding** ที่จะได้ความยาว 159 บิต ซึ่งได้แสดงการเปรียบเทียบไว้ในตารางที่ **2.5**

ตารางที่ 2.5 แสดงการเปรียบเทียบจำนวนโค้ดที่ได้

วิธีลดขนาดข้อมูล		ASCII		Huffman Coding		Adaptive Huffman	
ตัวอักษร	ความถี่	ขนาดโค้ด	จำนวนบิต	ขนาดโค้ด	จำนวนบิต	ขนาดโค้ด	จำนวนบิต
A	18	8	144	3	54	2	36
B	8	8	64	3	24	3	24
C	10	8	80	3	30	2	20
D	7	8	56	3	21	3	21
E	5	8	40	3	15	3	15
F	2	8	16	3	6	4	8
G	2	8	16	3	6	5	10
H	1	8	8	3	3	6	6
รวมขนาดข้อมูล (บิต)			**424**		**159**		**140**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วไปผลลัพธ์ของการของลดขนาดข้อมูลด้วยวิธีต่าง ๆ จะได้ประสิทธิภาพใกล้เคียงกัน ขึ้นอยู่กับลักษณะของข้อมูลนั้น ๆ วิธี **Adaptive Huffman Coding** นี้จะเหมาะกับข้อมูลที่มีตัวอักษรหลาย ๆ ตัวและมีความแตกต่างของข้อมูลมาก ๆ

ข้อมูลที่ทำการลดขนาดแล้วจะไม่สามารถกระทำได้ในระบบคอมพิวเตอร์ ดังนั้นเราจึงต้องนำโค้ดที่มีจำนวนบิตไม่เท่ากันมาเรียงต่อกันและนำกลับมาเข้ารหัสกลับเป็นรหัส **ASCII** โดยที่เราจะต้องสร้างโพรซีเยอร์ที่สามารถทำการถอดรหัสด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินการวิจัย

3.1 วิธีการดำเนินการวิจัย

การศึกษาและการดำเนินการวิจัยในปัญหาพิเศษฉบับนี้เป็นการวิจัยแบบทดลอง โดยทำการศึกษาทั้งทางด้านทฤษฎีและการประยุกต์ใช้วิธี **Adaptive Huffman Coding** จากหนังสือและวิทยานิพนธ์ที่เกี่ยวข้อง รวมทั้งการสร้างตัวอย่างโปรแกรม และวิเคราะห์ผลลัพธ์ในการลดขนาดข้อมูลของไฟล์ต่าง ๆ เช่น ไฟล์ข้อมูล **(Text Files)**,ไฟล์เอกสาร **(Document Files)** เป็นต้น

3.2 ระบบงาน

3.2.1 ลักษณะของข้อมูล **(Input)**

อาจเป็นข้อมูลที่อยู่ในรูปของโปรแกรมต่าง ๆ หรือไฟล์ประเภทข้อความ **(Text Files)**, ไฟล์เอกสาร **(Document Files)** หรือข้อมูลที่มีการจัดเก็บแบบชนิดตัวอักษร รวมทั้งไฟล์ประเภทอื่นๆด้วย

3.2.2 ส่วนวิเคราะห์และประมวลผลด้วยวิธี **Adaptive Huffman Coding**

- การเข้ารหัส **(Encode)** อ่านข้อมูลที่ต้องการจะลดขนาดทีละตัวนำมาวิเคราะห์และค้นหาตำแหน่งจัดเก็บที่เหมาะสมเพื่อจัดเก็บในแผนภูมิต้นไม้ **(Binary Tree)** โดยจะทำการเปลี่ยนแปลง **(Update)** แผนภูมิต้นไม้ตลอดเวลา แผนภูมิต้นไม้ที่ได้จะนำมาประมวลผลและกำหนดค่าข้อมูลใหม่แทนที่ข้อมูลเดิม

- การถอดรหัส **(Decode)** นำข้อมูลที่ถูกลดขนาดมาแปลงเป็นข้อมูลเดิม

3.2.3 ส่วนแสดงผล **(Output)**

หลังจากการประมวลผลจะได้ข้อมูลที่มีขนาดของข้อมูลน้อยลงจากเดิม และจะแสดงผลเป็นเปอร์เซ็นต์ที่สามารถทำการลดขนาดได้

3.3 ขั้นตอนการดำเนินงาน

3.3.1 ศึกษาปัญหาและวิธีการแก้ปัญหา

3.3.2 ศึกษาทฤษฎีที่เกี่ยวข้องกับ **Adaptive Huffman Coding**

3.3.3 ศึกษาแนวทางในการประยุกต์ใช้ **Adaptive Huffman Coding**

3.3.4 สร้างโปรแกรมตัวอย่างของการลดขนาดข้อมูลโดยวิธี **Adaptive Huffman Coding**

3.3.5 ทำการทดลองลดขนาดข้อมูลด้วยวิธี **Adaptive Huffman Coding** และวิธีอื่นๆ

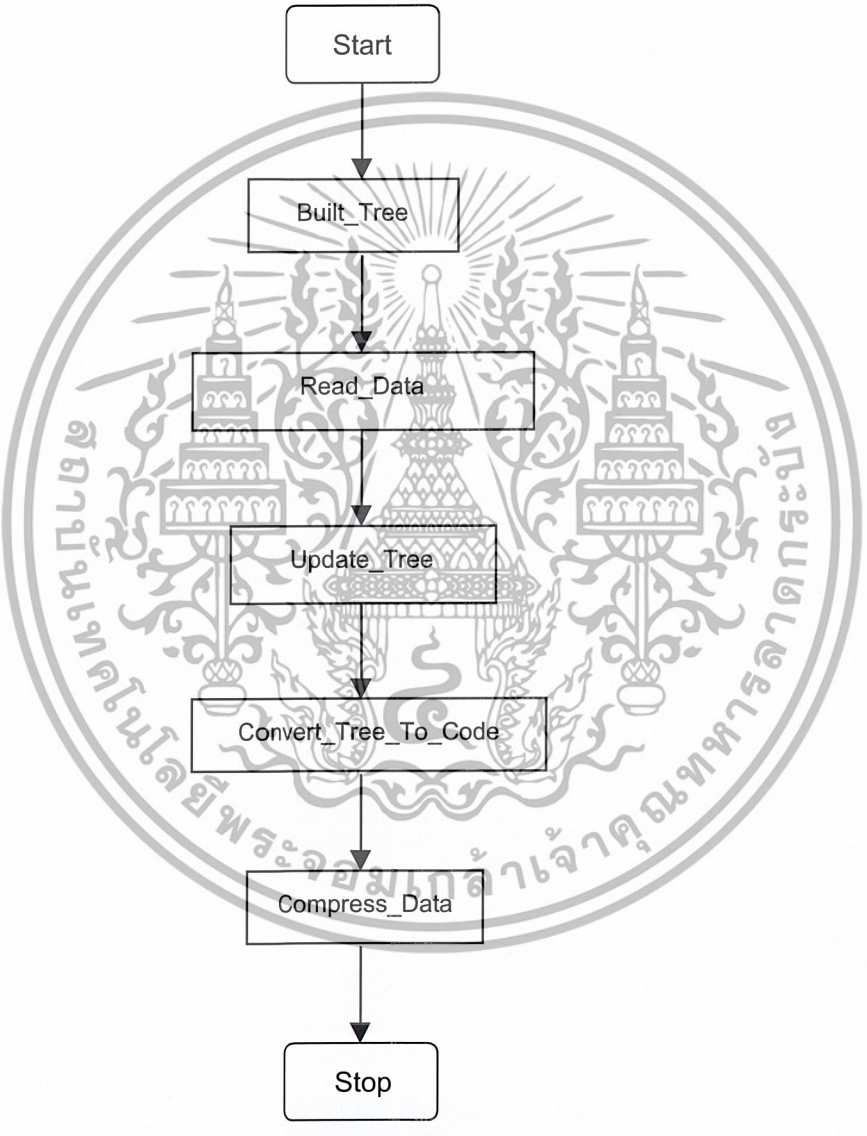
3.3.6 วิเคราะห์ผลลัพธ์ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.7 เขียนรายงานของการลดขนาดข้อมูลโดยวิธี **Adaptive Huffman Coding**

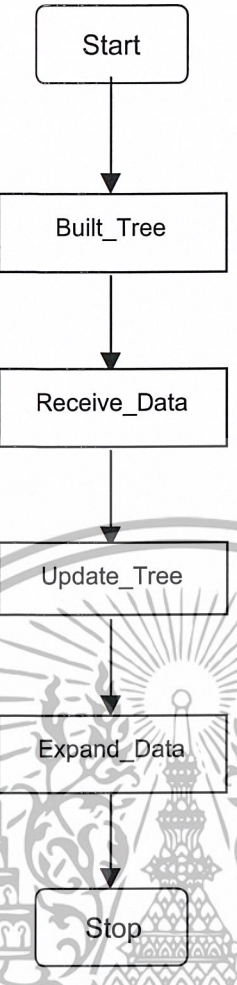
3.4 ขั้นตอนการทำงานของโปรแกรม Adaptive Huffman Coding

การสร้างโปรแกรมเพื่อใช้ลดขนาดข้อมูลด้วยวิธี **Adaptive Huffman Coding** ในปัญหาพิเศษฉบับนี้ เขียนขึ้นโดยใช้ภาษาซี โดยมีขั้นตอนการลดขนาดและขยายขนาดข้อมูลดังแผนผัง**รูปที่ 3.1** และ **รูปที่ 3.2**



รูปที่ 3.1 แผนผังแสดงขั้นตอนการลดขนาดไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนผังแสดงขั้นตอนการขยายข้อมูลให้กลับเป็นปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

รายงานผลการทดลอง

4.1 ขั้นตอนการทดลอง

โปรแกรม **Adaptive Huffman** นั้นเป็นโปรแกรมที่ต้องใช้ **RUN** บนระบบ **DOS** เท่านั้น ดังนั้นเมื่อผู้ใช้ต้องการใช้งานจึงต้องเข้าสู่ระบบ **DOS** ก่อน ขั้นตอนต่อมาคือเข้าไปสู่ไดเร็กทอรีที่เก็บโปรแกรม **Adaptive Huffman** ไว้แล้วจึงสามารถเริ่มใช้งานโปรแกรมได้ โดยจะต้องพิมพ์ตามนี้

[DIRECTORY]\comp.exe FILE.in FILE.out

ในส่วนของโปรแกรมที่ใช้ขยายขนาดก็ใช้งานเช่นเดียวกัน โดยที่จะต้องพิมพ์ตามนี้

[DIRECTORY]\uncom.exe FILE.in FILE.out

สำหรับตัวอย่างการใช้งานโปรแกรมสามารถดูได้ที่ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการวิเคราะห์การลดขนาดข้อมูลชนิดต่างๆด้วยวิธี Adaptive Huffman

4.2.1 ผลการทดลองลดขนาดข้อมูลชนิด DOC ด้วยวิธี Adaptive Huffman

ตารางที่ 4.2.1 เป็นการลดขนาดข้อมูลไฟล์ประเภท **DOC** ขนาดต่างๆกันจำนวนทั้งสิ้น 5 ไฟล์ โดยที่ประสิทธิภาพในการลดขนาดไม่ได้ขึ้นอยู่กับขนาดของข้อมูล แต่จะขึ้นกับความถี่ของการซ้ำกันของตัวอักษรในข้อมูล จากตัวอย่างทั้ง 5 ไฟล์ที่นำมาวิเคราะห์นั้นแสดงให้เห็นว่าการลดขนาดด้วยวิธี **Adaptive Huffman** สามารถลดขนาดข้อมูลได้เฉลี่ย 41.10% โดยประสิทธิภาพในการลดสูงสุดอยู่ที่ 44.67% และต่ำสุดอยู่ที่ 31.73% ส่วนการลดขนาดด้วยวิธี **Adaptive Huffman** ซ้ำกัน 2 ครั้งนั้นจะมีประสิทธิภาพในการลดขนาดเฉลี่ยอยู่ที่ 41.52%เมื่อเทียบกับขนาดเริ่มต้น เราจะเห็นได้ว่าประสิทธิภาพในการลดขนาดด้วยวิธี **Adaptive Huffman** ทั้งครั้งเดียวและ 2 ครั้งนั้นแทบจะไม่แตกต่างกัน ดังนั้นการลดขนาดซ้ำกัน 2 ครั้งนั้นจึงไม่เหมาะกับการลดขนาดไฟล์ **DOC** เพราะจะทำให้สิ้นเปลืองเวลาโดยไม่คุ้มค่า

ตารางที่ 4.2.1 แสดงผลการลดขนาดไฟล์ DOC ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	204800	122112	40.38	120611	41.11
2	384512	262527	31.73	260843	32.16
3	775680	420552	45.78	417916	46.12
4	212480	121237	42.94	119998	43.53
5	294400	162903	44.67	162905	44.67
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**41.10**		**41.52**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลองลดขนาดข้อมูลชนิด **TXT** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.2 เป็นผลการลดขนาดของไฟล์ประเภท **TXT** ขนาดต่างๆกันจำนวนทั้งสิ้น 5 ไฟล์ ซึ่งข้อมูลแต่ละชุดจะมีความถี่ของตัวอักษรแตกต่างกันไป จะเห็นว่าการลดขนาดไฟล์ประเภท **TXT** ด้วยวิธี **Adaptive Huffman** นั้นจะมีประสิทธิภาพเฉลี่ยประมาณ 26.18% โดยมีประสิทธิภาพสูงสุด 38.30% และต่ำสุด 9.43% ส่วนการลดขนาดซ้ำกัน 2 ครั้งนั้นจะมีประสิทธิภาพเฉลี่ยอยู่ที่ 31.98% โดยมีประสิทธิภาพสูงสุดคือ 41.77% และต่ำสุดคือ 23.26% จากผลการทดลองที่ได้ถึงแม้ว่าประสิทธิภาพเฉลี่ยของการลดขนาดซ้ำกัน 2 ครั้งนั้นจะดีกว่า แต่ถ้าเราตัดข้อมูลชุดที่ 5 ออก จะเห็นว่าประสิทธิภาพในการลดขนาดครั้งเดียวจะมีประสิทธิภาพดีกว่า

ตารางที่ 4.2.2 แสดงผลการลดขนาดไฟล์ **TXT** ด้วยวิธี **Adaptive Huffman**

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	20142	14498	28.02	14740	26.82
2	41190	28277	31.35	28539	30.71
3	27188	16774	38.30	17030	37.36
4	50310	38345	23.78	38608	23.26
5	14808	13411	9.43	8623	41.77
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**26.18**		**31.98**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดลองลดขนาดข้อมูลชนิด **JPG** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.3 เป็นการลดขนาดข้อมูลชนิดกราฟฟิกประเภท **JPG** ขนาดต่างๆกัน จำนวน 5 ไฟล์ ผลการลดขนาดไฟล์ทั้ง 5 นั้นได้ประสิทธิภาพเฉลี่ย –0.34% โดยข้อมูลทั้ง 5 หลังการลดขนาดจะมีขนาดที่ใหญ่ขึ้นทั้งหมด ส่วนการลดขนาดซ้ำกัน 2 ครั้งก็จะได้ผลลัพธ์ที่ใกล้กันคือ –0.52% ทั้งนี้เพราะไฟล์ประเภท **JPG** เป็นไฟล์ภาพที่ได้ผ่านการลดขนาดมาก่อนแล้ว ดังนั้นวิธี **Adaptive Huffman** จึงไม่เหมาะจะนำมาลดขนาดไฟล์ **JPG**

ตารางที่ 4.2.3 แสดงผลการลดขนาดไฟล์ JPG ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	109330	109476	-0.13	109768	-0.40
2	71265	71423	-0.22	71702	-0.61
3	113014	113288	-0.24	113565	-0.48
4	109791	109801	0.00	110101	-0.28
5	63134	63391	-0.40	63663	-0.83
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**-0.19**		**-0.52**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 ผลการทดลองลดขนาดข้อมูลชนิด **GIF** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.4 เป็นผลการลดขนาดไฟล์ประเภท **GIF** ขนาดต่างๆกันจำนวนทั้งสิ้น 5 ไฟล์ซึ่งไฟล์ประเภท **GIF** ก็เป็นไฟล์กราฟฟิกที่ได้ผ่านการลดขนาดมาก่อนแล้วเหมือนกัน จากการทดลองลดขนาดด้วยวิธี **Adaptive Huffman** ทั้งครั้งเดียวและ 2 ครั้ง ได้ประสิทธิภาพเฉลี่ย 0.42% และ 1.84% ตามลำดับ เราสามารถสรุปได้ว่าไฟล์ **GIF** ไม่เหมาะจะนำมาลดขนาดด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.4 แสดงผลการลดขนาดไฟล์ GIF ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	45006	45092	-0.19	45365	-0.79
2	97915	97747	0.17	98022	-0.10
3	103582	101108	2.39	101392	2.11
4	157491	157626	0.00	157960	-0.26
5	394084	361388	8.30	361677	8.22
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**2.13**		**1.84**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 ผลการทดลองลดขนาดข้อมูลชนิด **WAV** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.5 เป็นการลดขนาดไฟล์เสียงประเภท **WAV** ขนาดต่างๆกันทั้งสิ้น จำนวน 5 ไฟล์ จากผลการทดลองลดขนาด วิธี **Adaptive Huffman** สามารถลดขนาดข้อมูลได้เฉลี่ย 13.87% โดยที่ประสิทธิภาพสูงสุดอยู่ที่ 21.64% และมีประสิทธิภาพต่ำสุดอยู่ที่ 7.91% ส่วนการลดขนาดซ้ำกัน 2 ครั้งนั้นได้ผลลัพธ์ที่ใกล้เคียงกับการลดขนาดครั้งเดียวโดยมีประสิทธิภาพเฉลี่ย 13.52% ดังนั้นการลดขนาดซ้ำ 2 ครั้งจึงไม่เหมาะสม

ตารางที่ 4.2.5 แสดงผลการลดขนาดไฟล์ WAV ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	97016	76018	21.64	76300	21.35
2	23430	20076	14.31	20318	13.28
3	313116	288351	7.91	288630	7.82
4	119384	100264	16.02	100559	15.77
5	693212	627730	9.45	628060	9.40
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**13.87**		**13.52**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 ผลการทดลองลดขนาดข้อมูลชนิด **MP3** ด้วยวิธี **Adaptive Huffman**

ตารางที่4.2.6 เป็นผลของการลดขนาดไฟล์เสียงประเภท **MP3** จำนวน 5 ไฟล์ โดยผลการทดลองที่ได้นั้นมีประสิทธิภาพเฉลี่ยอยู่ที่ 0.48% ส่วนการลดขนาดซ้ำกัน 2 ครั้งด้วยวิธี **Adaptive Huffman** นั้นได้ประสิทธิภาพเฉลี่ย 0.47% ซึ่งแทบจะไม่แตกต่างกันเลย จากผลการทดลองที่ได้จะเห็นว่าประสิทธิภาพในการลดขนาดข้อมูลทุกชุดมีประสิทธิภาพน้อยและใกล้เคียงกันทั้งหมด เพราะไฟล์ **MP3** เป็นไฟล์เสียงที่ผ่านการลดขนาดมาแล้วนั่นเอง

ตารางที่ 4.2.6 แสดงผลการลดขนาดไฟล์ MP3 ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	3606560	3589041	0.49	3589324	0.48
2	3705093	3687119	0.49	3687398	0.48
3	3525063	3509664	0.44	3509936	0.43
4	3396754	3376711	0.59	3376999	0.58
5	3180658	3168354	0.39	3168632	0.39
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**0.48**		**0.47**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 ผลการทดลองลดขนาดข้อมูลชนิด **EXE** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.7 เป็นการลดขนาดข้อมูลชนิดบิตที่มีนามสกุล **EXE** ขนาดต่างๆกัน จำนวน 5 ไฟล์ โดยที่ประสิทธิภาพในการลดขนาดนั้นจะขึ้นกันเนื้อหาข้อมูลหรือความถี่ของโครงสร้างข้อมูล ผลลัพธ์หลังจากการลดขนาดจะได้ประสิทธิภาพเฉลี่ย 14.29% โดยมีประสิทธิภาพสูงสุดที่ 20.71% และต่ำสุด 0.16% ส่วนการลดขนาดด้วยวิธี **Adaptive Huffman** 2 ครั้งได้ประสิทธิภาพเฉลี่ย 14.39% ซึ่งมีประสิทธิภาพใกล้เคียงกัน

ตารางที่ 4.2.7 แสดงผลการลดขนาดไฟล์ EXE ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	3613871	3608218	0.16	3608603	0.15
2	6561792	5529433	15.73	5528462	15.75
3	13828096	11367994	17.79	11326700	18.09
4	2054213	1628806	20.71	1626962	20.80
5	4194304	3477971	17.08	3475390	17.14
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**14.29**		**14.39**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 ผลการทดลองลดขนาดข้อมูลชนิด **DAT** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.8 เป็นการลดขนาดข้อมูลประเภท **DAT** ซึ่งเป็นไฟล์ชนิดภาพเคลื่อนไหวจำนวนทั้งสิ้น 5 ไฟล์ โดยมีประสิทธิภาพเฉลี่ยอยู่ที่ 29.73% มีประสิทธิภาพสูงสุด 62.69% และประสิทธิภาพต่ำสุด 9.72% ส่วนการลดขนาดด้วยวิธี **Adaptive Huffman** ซ้ำกัน 2 ครั้งจะมีประสิทธิภาพเฉลี่ย 32.48% โดยมีประสิทธิภาพสูงสุดคือ 73.87% และประสิทธิภาพต่ำสุดคือ 9.63%

ตารางที่ 4.2.8 แสดงผลการลดขนาดไฟล์ DAT ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	311250	280988	9.72	281265	9.63
2	614147	477064	22.32	475929	22.51
3	732116	570991	22.01	570120	22.13
4	2007040	1372282	31.63	1318914	34.28
5	5718016	2118106	62.96	1714060	73.87
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**29.73**		**32.48**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.9 ผลการทดลองลดขนาดข้อมูลชนิด **WMV** ด้วยวิธี **Adaptive Huffman**

ตารางที่ 4.2.8 เป็นการลดขนาดข้อมูลประเภท **WMV** ซึ่งเป็นไฟล์ชนิดภาพเคลื่อนไหวจำนวนทั้งสิ้น 5 ไฟล์ โดยมีประสิทธิภาพเฉลี่ยอยู่ที่ 0.64% มีประสิทธิภาพสูงสุด 0.90% และประสิทธิภาพต่ำสุด 0.33% เท่านั้นส่วนการลดขนาดด้วยวิธี **Adaptive Huffman** ซ้ำกัน 2 ครั้งจะมีประสิทธิภาพเฉลี่ยที่น้อยลงกว่าเดิมคือ 0.62% โดยมีประสิทธิภาพสูงสุดคือ 0.89% และประสิทธิภาพต่ำสุดคือ 0.32%

ตารางที่ 4.2.9 แสดงผลการลดขนาดไฟล์ WMV ด้วยวิธี Adaptive Huffman

ข้อมูลชุดที่	ขนาดเริ่มต้น	ขนาดหลังการลดขนาดครั้งแรก	เปอร์เซ็นต์การลด	ขนาดหลังการลดขนาดครั้งที่ 2	เปอร์เซ็นต์การลด
1	1907023	1892570	0.76	1892903	0.74
2	818893	812491	0.78	812785	0.75
3	2215931	2206655	0.42	2206962	0.40
4	2537113	2528632	0.33	2528946	0.32
5	3001157	2974060	0.90	2974348	0.89
เปอร์เซ็นต์การลดขนาดเฉลี่ย			**0.64**		**0.62**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการวิเคราะห์ผลการลดขนาดครั้งที่ 2 ด้วยวิธีต่างๆหลังจากลดขนาดครั้งแรกด้วย Adaptive Huffman

4.3.1 ผลการลดขนาดข้อมูลชนิด **DOC** ครั้งที่ 2

จาก**ตารางที่ 4.3.1** เราจะเห็นได้ว่าการลดขนาดในครั้งแรกด้วยวิธี **Adaptive Huffman** ยังไม่ให้ผลลัพธ์ที่ดีพอเพราะเมื่อเราทำการลดขนาดครั้งที่ 2 ด้วยวิธีต่างๆ จะเห็นว่าวิธี **WINZIP Adaptive Huffman** รวมไปถึง **LZW** ยังสามารถทำการลดขนาดลงไปได้อีก ดังนั้นเราอาจสรุปได้ว่า วิธี **Adaptive Huffman** ยังไม่ใช่วิธีที่ดีที่สุดในการลดขนาดไฟล์ประเภท **DOC** ก็ได้

ตารางที่ 4.3.1 ตารางแสดงผลการลดขนาดไฟล์ประเภท DOC ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
204800	40.38	105230	48.62	120611	41.11	117105	42.82	139378	31.94
384512	31.73	223792	41.80	260843	32.16	251219	34.67	291326	24.23
775680	45.78	348197	55.11	417916	46.12	411374	46.97	469527	39.47
212480	42.94	106807	49.73	119998	43.53	116609	45.12	136957	35.54
294400	44.67	157086	46.64	162905	44.67	163775	44.37	186708	36.58
เปอร์เซ็นต์การลดเฉลี่ย	**41.10**		**48.38**		**41.52**		**42.79**		**33.55**

4.3.2 ผลการลดขนาดข้อมูลชนิด **TXT** ครั้งที่ 2

จาก**ตารางที่ 4.3.2** เราจะเห็นได้ว่าการลดขนาดในครั้งแรกด้วยวิธี **Adaptive Huffman** สำหรับไฟล์ประเภท **TXT** ยังให้ผลลัพธ์ที่ไม่ดีพอเพราะเมื่อเราทำการลดขนาดครั้งที่ 2 ด้วยวิธีต่างๆ จะเห็นว่าวิธี **WINZIP Adaptive Huffman** รวมไปถึง **LZW** ยังสามารถทำการลดขนาดลงไปได้อีก โดยเฉพาะวิธี **WINZIP** สามารถลดขนาดได้เพิ่มอีกถึง 14% ดังนั้นเราอาจสรุปได้ว่า วิธี **Adaptive Huffman** ยังไม่ใช่วิธีที่ดีที่สุดในการลดขนาดไฟล์ประเภท **TXT** ก็ได้

ตารางที่ 4.3.2 ตารางแสดงผลการลดขนาดไฟล์ประเภท TXT ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
20142	28.02	14345	28.78	14740	26.82	14730	26.87	18565	7.83
41190	31.35	28192	31.56	28539	30.71	28767	30.16	32360	21.44
27188	38.30	16490	39.35	17030	37.36	17037	37.34	20855	23.29
50310	23.78	38020	24.43	38608	23.26	39048	22.38	46501	7.57
14808	9.43	3081	79.19	8623	41.77	6417	56.66	11684	21.10
เปอร์เซ็นต์การลดเฉลี่ย	**26.18**		**40.66**		**31.98**		**34.68**		**16.25**

4.3.3 ผลการลดขนาดข้อมูลชนิด **JPG** ครั้งที่ 2

จาก**ตารางที่ 4.3.3** ไฟล์ **JPG** นั้นเป็นไฟล์ข้อมูลที่ถูกบีบอัดมาก่อนแล้ว และในการลดขนาดครั้งแรกก็ได้ขนาดที่เพิ่มขึ้น ซึ่งหมายความว่าจะได้โค้ดที่มีความแตกต่างกันมากขึ้นด้วย ดังนั้นในการลดขนาดครั้งที่ 2 ไม่ว่าจะด้วยวิธีใดก็ตามจึงให้ผลลัพธ์ไม่น่าพอใจเลย โดยเฉพาะวิธี **Huffman** ที่ให้ผลลัพธ์มีขนาดเพิ่มขึ้นถึง 15%

ตารางที่ 4.3.3 ตารางแสดงผลการลดขนาดไฟล์ประเภท JPG ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
109330	-0.13	109610	-0.25	109768	-0.40	111585	-2.06	125840	-15.10
71265	-0.22	71552	-0.40	71702	-0.61	72851	-2.22	83683	-17.42
113014	-0.24	113422	-0.36	113565	-0.48	115466	-2.16	129646	-14.71
109791	0.00	109865	-0.06	110101	-0.28	111912	-1.93	126147	-14.89
63134	-0.40	63515	-0.60	63663	-0.83	64705	-2.48	75638	-19.80
เปอร์เซ็นต์การลดเฉลี่ย	**-0.19**		**-0.33**		**-0.52**		**-2.17**		**-16.38**

4.3.4 ผลการลดขนาดข้อมูลชนิด **GIF** ครั้งที่ 2

จาก**ตารางที่ 4.3.4** เนื่องจากไฟล์ประเภท **GIF** ก็เป็นไฟล์กราฟฟิกที่ถูกบีบอัดมาแล้วเหมือนกันกับไฟล์ประเภท **JPG** ดังนั้นผลลัพธ์ในการลดขนาดครั้งที่ 2 จึงให้ผลลัพธ์ที่ไม่แตกต่างกันมากนัก โดยประสิทธิภาพในการลดขนาดครั้งที่ 2 จะมีประสิทธิภาพด้อยกว่าการลดขนาดครั้งแรกเพียงเล็กน้อยเท่านั้น

ตารางที่ 4.3.4 ตารางแสดงผลการลดขนาดไฟล์ประเภท GIF ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
45006	-0.19	45216	-0.46	45365	-0.79	46049	-2.31	53277	-18.37
97915	0.17	97876	0.04	98022	-0.10	99662	-1.78	114085	-16.51
103582	2.39	101242	2.26	101392	2.11	103111	0.45	117464	-13.40
157491	0.00	157765	-0.17	157960	-0.26	160612	-1.98	182174	-15.67
394084	8.30	361547	8.25	361677	8.22	367719	6.69	414565	-5.19
เปอร์เซ็นต์การลดเฉลี่ย	**2.13**		**1.98**		**1.84**		**0.21**		**-13.82**

4.3.5 ผลการลดขนาดข้อมูลชนิด **WAV** ครั้งที่ 2

จาก**ตารางที่ 4.3.5** เมื่อเราทำการลดขนาดครั้งที่ 2 ด้วยวิธี **WINZIP** นั้นสามารถลดขนาดได้เพิ่มขึ้นอีกเพียงเล็กน้อยเท่านั้น โดยที่การลดขนาดครั้งที่ 2 ด้วยวิธีอื่นๆนั้นได้ผลลัพธ์ที่มีขนาดเพิ่มขึ้นเล็กน้อย ยกเว้นแต่วิธี **Huffman** ที่ให้ผลลัพธ์ที่มีขนาดเพิ่มมากขึ้นอีกถึง 10%

ตารางที่ 4.3.5 ตารางแสดงผลการลดขนาดไฟล์ประเภท WAV ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
97016	21.64	76005	21.66	76300	21.35	77376	20.24	86423	10.92
23430	14.31	20046	14.44	20318	13.28	20329	13.23	23070	1.54
313116	7.91	288453	7.88	288630	7.82	293427	6.29	319522	-2.04
119384	16.02	100255	16.02	100559	15.77	102144	14.44	110549	7.40
693212	9.45	626525	9.62	628060	9.40	639153	7.80	674481	2.70
เปอร์เซ็นต์การลดเฉลี่ย	**13.87**		**13.92**		**13.52**		**12.40**		**4.10**

4.3.6 ผลการลดขนาดข้อมูลชนิด **MP3** ครั้งที่ 2

จาก**ตารางที่ 4.3.6** ไฟล์ MP3 นั้นเป็นไฟล์เสียงที่ผ่านการบีบอัดมาก่อนแล้ว ดังนั้นประสิทธิภาพในการลดขนาดทั้งครั้งที่ 1 และครั้งที่ 2 จึงให้ผลลัพธ์ที่ไม่ดีนัก โดยที่วิธี **WINZIP** สามารถลดขนาดได้มากกว่าครั้งแรกอีกหนึ่งเท่า ส่วนวิธี **LZW** และ **Huffman** จะให้ผลลัพธ์ที่มีขนาดเพิ่มขึ้น

ตารางที่ 4.3.6 ตารางแสดงผลการลดขนาดไฟล์ประเภท MP3 ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
3606560	0.49	3565214	1.15	3589324	0.48	3641045	-0.95	4072572	-12.92
3705093	0.49	3663557	1.12	3687398	0.48	3744706	-1.06	4185378	-12.96
3525063	0.44	3486803	1.08	3509936	0.43	3566001	-1.16	3982778	-12.98
3396754	0.59	3352718	1.30	3376999	0.58	3430502	-0.99	3828932	-12.72
3180658	0.39	3144845	1.12	3168632	0.39	3218602	-1.19	3597277	-13.09
เปอร์เซ็นต์การลดเฉลี่ย	**0.48**		**1.15**		**0.47**		**-1.07**		**-12.93**

4.3.7 ผลการลดขนาดข้อมูลชนิด **EXE** ครั้งที่ 2

จาก**ตารางที่ 4.3.7** จะเห็นได้ว่าวิธี **WINZIP** สามารถลดขนาดได้มากกว่าครั้งแรกอีกถึง 8% และวิธี **Adaptive Huffman** และ **LZW** ให้ผลลัพธ์ที่ใกล้เคียงกับการลดขนาดครั้งแรก ดังนั้นเราอาจจะสรุปได้ว่าวิธี **Adaptive Huffman** ไม่สามารถทำการลดขนาดไฟล์ประเภท **EXE** ได้ละเอียดเพียงพอ

ตารางที่ 4.3.7 ตารางแสดงผลการลดขนาดไฟล์ประเภท EXE ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
3613871	0.16	3594964	0.52	3608603	0.15	3671035	-1.58	4097458	-13.38
6561792	15.73	4578280	30.23	5528462	15.75	5608287	14.53	6219141	5.22
13828096	17.79	9439642	31.73	11326700	18.09	11383846	17.67	12623172	8.71
2054213	20.71	1439516	29.92	1626962	20.80	1653457	19.51	1825327	11.14
4194304	17.08	3293290	21.48	3475390	17.14	3520131	16.07	3918962	6.65
เปอร์เซ็นต์การลดเฉลี่ย	**14.29**		**22.78**		**14.39**		**13.24**		**3.67**

4.3.8 ผลการลดขนาดข้อมูลชนิด **DAT** ครั้งที่ 2

จาก**ตารางที่ 4.3.8** เราจะเห็นว่าในการลดขนาดครั้งที่ 2 ให้ผลลัพธ์ที่มีประสิทธิภาพที่ดีมากขึ้นกว่าครั้งแรกยกเว้น**วิธี Huffman** โดยเฉพาะในข้อมูลตัวที่ 5 ดังนั้นเราอาจสรุปได้ว่าวิธี **Adaptive Huffman** ไม่สามารถทำการลดขนาดไฟล์ประเภท **DAT** ได้ละเอียดมากพอ

ตารางที่ 4.3.8 ตารางแสดงผลการลดขนาดไฟล์ประเภท DAT ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
311250	9.72	277817	10.74	281265	9.63	285614	8.24	321873	-3.41
614147	22.32	447959	27.06	475929	22.51	479031	22.00	539240	12.20
732116	22.01	532244	27.30	570120	22.13	578400	21.00	645355	11.85
2007040	31.63	969931	51.67	1318914	34.28	1269218	36.76	1416156	29.44
5718016	62.96	401511	92.98	1714060	73.87	2110660	63.09	1494281	73.87
เปอร์เซ็นต์การลดเฉลี่ย	**29.73**		**41.95**		**32.48**		**30.22**		**24.79**

4.3.9 ผลการลดขนาดข้อมูลชนิด **WMV** ครั้งที่ 2

จาก**ตารางที่ 4.3.9** ผลการลดขนาดครั้งแรกด้วย **Adaptive Huffman** และลดขนาดครั้งที่ 2 ด้วย **WINZIP** นั้นให้ผลลัพธ์ที่ไม่แตกต่างกันมากนัก ส่วนวิธีอื่นให้ผลลัพธ์ที่มีประสิทธิภาพด้อยลงกว่าเดิม

ตารางที่ 4.3.9 ตารางแสดงผลการลดขนาดไฟล์ประเภท WMV ครั้งที่ 2 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	เปอร์เซ็นต์การลดครั้งแรก	WINZIP		ADAPTIVE HUFFMAN		LZW		HUFFMAN	
		ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด	ขนาด(Bytes)	%การลด
1907023	0.76	1878388	1.50	1892903	0.74	1917032	-0.52	2152998	-12.89
818893	0.78	811704	0.88	812785	0.75	826097	-0.87	927008	-13.20
2215931	0.42	2197980	0.81	2206962	0.40	2242648	-1.20	2509318	-13.23
2537113	0.33	2525252	0.47	2528946	0.32	2567535	-1.19	2876251	-13.36
3001157	0.90	2963927	1.24	2974348	0.89	3023541	-0.74	3383304	-12.73
เปอร์เซ็นต์การลดเฉลี่ย	**0.64**		**0.98**		**0.62**		**-0.90**		**-13.08**

4.4 ผลการทดลองด้วยวิธี Adaptive Huffman เปรียบเทียบกับวิธีอื่นๆ

4.4.1 ผลการทดลองการลดขนาดข้อมูลชนิด **DOC** ด้วยวิธีต่างๆ

ตารางที่ 4.4.1 เป็นตารางแสดงผลการทดลองการลดขนาดข้อมูลประเภท **DOC** ด้วยวิธีต่างๆ จากตารางเปรียบเทียบจะเห็นว่า วิธี **Adaptive Huffman** มีประสิทธิภาพที่เหนือกว่าวิธี **LZW** และ **Huffman** โดยที่วิธี **Adaptive Huffman** จะมีประสิทธิภาพในการลดขนาดเฉลี่ยอยู่ที่ 41.10% ส่วนวิธี **Huffman** มีประสิทธิภาพในการลดขนาดเฉลี่ย 35.30% เท่านั้น แต่วิธีที่มีประสิทธิภาพดีที่สุดคือวิธี **WINZIP** ซึ่งมีประสิทธิภาพเฉลี่ย 78.83%

ตารางที่4.4.1 แสดงผลการลดขนาดไฟล์ DOC ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
204800	41806	2.55	79.59	122112	0.94	40.38	122742	0.97	40.07	132214	0.16	35.44
384512	53420	2.56	81.22	262527	1.49	31.73	254717	1.58	33.75	264614	0.50	31.18
775680	165055	2.72	78.72	420552	2.03	45.78	393314	1.99	49.29	468549	1.32	39.60
212480	49561	2.62	76.67	121237	0.99	42.94	118174	1.03	44.38	130231	0.33	38.71
294400	64925	2.75	77.95	162903	1.12	44.67	168114	1.46	42.89	201425	0.44	31.58
เปอร์เซ็นต์การลดเฉลี่ย			**78.83**			**41.10**			**42.08**			**35.30**

4.4.2 ผลการทดลองการลดขนาดข้อมูลชนิด **TXT** ด้วยวิธีต่างๆ

ตารางที่ 4.4.2 เป็นตารางแสดงการเปรียบเทียบผลลัพธ์ของการลดขนาดข้อมูลชนิด **TXT** ด้วยวิธีต่างๆ เราจะเห็นได้ว่าวิธี **WINZIP** มีประสิทธิภาพสูงที่สุดคือ 68.46% ส่วนวิธี **Huffman** และ **LZW** จะมีประสิทธิภาพในการลดขนาดน้อยที่สุดคือ 17.30% และ 16.59% ตามลำดับ ถ้าเราทำการลดขนาดซ้ำ 2 ครั้งด้วยวิธี **Huffman** จะได้ประสิทธิภพที่ต่ำลงไปอีกคือ 1.12% เท่านั้น

ตารางที่4.4.2 แสดงผลการลดขนาดไฟล์ TXT ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
20142	8325	2.57	58.67	14498	0.30	28.02	17693	0.29	12.16	16881	0.02	16.19
41190	18951	2.50	53.99	28277	0.51	31.35	35676	0.52	13.39	31348	0.06	23.89
27188	9078	2.50	66.61	16774	0.46	38.30	21930	0.55	19.34	18095	0.00	33.44
50310	16583	2.53	67.03	38345	0.52	23.78	43681	0.47	13.18	42433	0.05	15.66
14808	558	2.48	96.23	13411	0.43	9.43	11124	0.43	24.88	15204	0.00	-2.67
เปอร์เซ็นต์การลดเฉลี่ย			**68.46**			**26.18**			**16.59**			**17.30**

4.4.3 ผลการทดลองการลดขนาดข้อมูลชนิด **JPG** ด้วยวิธีต่างๆ

ตารางที่ 4.4.3 เป็นตารางแสดงผลการทดลองการลดขนาดไฟล์ข้อมูลกราฟฟิกประเภท **JPG** เนื่องจากไฟล์ประเภท **JPG** เป็นไฟล์กราฟฟิกที่ได้ถูกทำการลดขนาดมาก่อนแล้วจึงทำให้ได้ประสิทธิภาพในการลดขนาดต่ำทุกวิธี โดยถ้าทำการลดขนาดด้วยวิธี **Adaptive Huffman** จะมีขนาดเพิ่มขึ้น 0.34% และถ้าใช้วิธี **Huffman** จะมีขนาดเพิ่มขึ้นมากถึง 15.92% โดยวิธีที่ดีที่สุดคือ **WINZIP** ซึ่งสามารถลดขนาดได้เพียงแค่ 2.80% เท่านั้น

ตารางที่4.4.3 แสดงผลการลดขนาดไฟล์ JPG ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
109330	108636	2.56	0.63	109476	0.85	-0.13	110522	0.92	-1.09	125328	0.16	-14.63
71265	64523	2.50	9.46	71423	0.77	-0.22	70127	0.75	1.60	83216	0.11	-16.77
113014	110364	2.59	2.34	113288	0.89	-0.24	114698	0.88	-1.49	129307	0.16	-14.41
109791	108203	2.52	1.45	109801	0.94	-0.01	110811	0.94	-0.92	125621	0.22	-14.42
63134	63056	2.54	0.12	63391	0.72	-0.40	64012	0.76	-1.39	75353	0.11	-19.35
เปอร์เซ็นต์การลดเฉลี่ย			**2.80**			**-0.19**			**-0.65**			**-15.92**

4.4.4 ผลการทดลองการลดขนาดข้อมูลชนิด **GIF** ด้วยวิธีต่างๆ

ตารางที่ 4.4.4 เป็นตารางแสดงผลการลดขนาดข้อมูลกราฟฟิกประเภท **GIF** ซึ่งเป็นไฟล์กราฟฟิกที่ได้ผ่านการลดขนาดมาก่อนแล้วเหมือนกันแต่คนละวิธีกับ **JPG** โดยวิธีที่มีประสิทธิภาพสูงสุดคือ **WINZIP** ซึ่งสามารถลดขนาดได้ถึง 29.43% ส่วนวิธี **Adaptive Huffman** สามารถขนาดได้ 2.13% แต่ถ้าเราใช้วิธี **Huffman** เราจะได้ขนาดหลังการลดมีขนาดเพิ่มขึ้นมากถึง 13.39% ซึ่งจะทำให้เปลืองเนื้อที่จัดเก็บมากขึ้น

ตารางที่4.4.4 แสดงผลการลดขนาดไฟล์ GIF ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
45006	35813	2.54	20.43	45092	0.46	-0.19	45647	0.50	-1.42	52958	0.06	-17.67
97915	16689	2.53	82.96	97747	0.72	0.17	98158	0.76	-0.24	113660	0.11	-16.08
103582	100837	2.53	2.65	101108	0.83	2.39	102569	0.88	0.98	116840	0.11	-12.80
157491	157555	2.60	0.00	157626	0.97	-0.01	159882	0.95	-1.51	181718	0.22	-15.38
394084	231966	2.67	41.13	361388	1.64	8.30	364504	1.73	7.51	413811	0.38	-5.01
เปอร์เซ็นต์การลดเฉลี่ย			**29.43**			**2.13**			**1.06**			**-13.39**

4.4.5 ผลการทดลองการลดขนาดข้อมูลชนิด **WAV** ด้วยวิธีต่างๆ

ตารางที่ 4.4.5 เป็นตารางแสดงการเปรียบเทียบผลการทดลองการลดขนาดไฟล์เสียงชนิด **WAV** ด้วยวิธีต่างๆ โดย**วิธีที่มีประสิทธิภาพดีที่สุดคือ WINZIP** ซึ่งมีประสิทธิภาพสูงใกล้เคียงกับวิธีรองลงมาคือวิธี **Adaptive Huffman** ซึ่งมีประสิทธิภาพ 18.96% และ 13.87% เรียงลงมาตามลำดับ ส่วนวิธี **Huffman** สามารถขนาดได้เพียงแค่ 1.69% เท่านั้น

ตารางที่4.4.5 แสดงผลการลดขนาดไฟล์ WAV ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
97016	74372	2.63	23.34	76018	0.76	21.64	84378	0.82	12.03	89378	0.05	7.87
23430	17386	2.58	25.79	20076	0.60	14.31	21014	0.62	10.31	23601	0.00	-0.72
313116	283116	2.82	9.58	288351	1.47	7.91	306940	1.47	1.97	326873	0.28	-4.39
119384	97068	2.63	18.69	100264	0.86	16.02	109259	0.88	8.48	113150	0.22	5.22
693212	572586	3.27	17.40	627730	2.78	9.45	667301	3.01	3.74	689788	0.61	0.49
เปอร์เซ็นต์การลดเฉลี่ย			**18.96**			**13.87**			**7.31**			**1.69**

4.4.6 ผลการทดลองการลดขนาดข้อมูลชนิด **MP3** ด้วยวิธีต่างๆ

ตารางที่ 4.4.6 เป็นตารางแสดงผลลัพธ์การลดขนาดไฟล์เสียงประเภท **MP3** โดยที่ไฟล์ **MP3** เป็นไฟล์เสียงที่ได้ผ่านการลดขนาดมาก่อนแล้ว ทำให้การลดขนาดด้วยวิธีต่างๆมีประสิทธิภาพไม่ดีนัก โดยวิธีที่มีประสิทธิภาพมากที่สุดคือ **WINZIP** ซึ่งสามารถลดขนาดลงมาได้เพียง 1.76% ส่วนวิธี **Adaptive Huffman** นั้นจะสามารถลดขนาดได้ 0.48% ส่วนวิธี **LZW** จะสามารถลดขนาดได้แค่ 0.23% เท่านั้น ทั้งนี้โปรแกรม **Huffman** ที่ใช้ไม่สามารถทำการลดขนาดไฟล์ **MP3** ได้

ตารางที่4.4.6 แสดงผลการลดขนาดไฟล์ MP3 ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
3606560	3543061	6.22	1.76	3589041	13.78	0.49	3595595	14.26	0.30			
3705093	3642230	8.24	1.69	3687119	15.89	0.49	3701872	16.14	0.09			
3525063	3466926	8.17	1.65	3509664	15.23	0.44	3517426	15.33	0.22			
3396754	3326750	7.95	2.06	3376711	14.67	0.59	3381473	14.71	0.45			
3180658	3127645	5.70	1.66	3168354	12.64	0.39	3177565	12.84	0.10			
เปอร์เซ็นต์การลดเฉลี่ย			**1.76**			**0.48**			**0.23**			

โปรแกรม HUFFMAN ไม่สามารถทำการลดขนาดไฟล์ MP3 ได้

4.4.7 ผลการทดลองการลดขนาดข้อมูลชนิด **EXE** ด้วยวิธีต่างๆ

ตารางที่ 4.4.7 เป็นตารางแสดงผลการลดขนาดข้อมูลประเภท **EXE** ด้วยวิธีต่างๆ จะเห็นว่าวิธี **Adaptive Huffman** และ **Huffman** จะมีประสิทธิภาพในการลดขนาดต่างกันถึงเท่าตัวคือ 14.29% และ 7.82% ตามลำดับ ส่วนวิธี **WINZIP** จะสามารถให้ผลลัพธ์ที่ดีที่สุดคือ 44.15%

ตารางที่4.4.7 แสดงผลการลดขนาดไฟล์ EXE ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
3613871	3466278	8.03	4.08	3608218	14.98	0.16	3764186	16.12	-4.15	3949306	7.25	-9.28
6561792	2530653	10.02	61.43	5529433	21.34	15.73	5659203	22.57	13.75	5760132	10.71	12.22
13828096	5919121	18.91	57.19	11367994	48.76	17.79	11356491	43.69	17.87	11937600	22.41	13.67
2054213	851002	4.83	58.57	1628806	7.75	20.71	1694868	7.84	17.49	1712271	3.68	16.65
4194304	2538106	8.24	39.49	3477971	13.62	17.08	3604480	13.95	14.06	3949306	7.14	5.84
เปอร์เซ็นต์การลดเฉลี่ย			**44.15**			**14.29**			**11.80**			**7.82**

4.4.8 ผลการทดลองการลดขนาดข้อมูลชนิด **DAT** ด้วยวิธีต่างๆ

ตารางที่ 4.4.8 เป็นตารางแสดงผลลัพธ์จากการลดขนาดไฟล์ข้อมูลภาพเคลื่อนไหวประเภท **DAT** ด้วยวิธีต่างๆ โดยที่วิธี **Adaptive Huffman LZW** และ **Huffman** จะมีประสิทธิภาพในการลดขนาดใกล้เคียงกันคือ 29.73% 28.63% และ 27.59% ตามลำดับโดยวิธีที่ให้ผลลัพธ์ที่ดีที่สุดคือวิธี **WINZIP** ซึ่งให้ผลลัพธ์คือ 60.11%

ตารางที่4.4.8 แสดงผลการลดขนาดไฟล์ DAT ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
311250	216304	2.16	30.50	280988	1.53	9.72	294651	1.72	5.33	321961	0.27	-3.44
614147	318112	2.73	48.20	477064	2.18	22.32	452439	2.01	26.33	537754	0.61	12.44
732116	335605	2.85	54.16	570991	2.52	22.01	643383	3.43	12.12	557979	0.55	23.79
2007040	584452	3.09	70.88	1372282	5.49	31.63	1756812	6.87	12.47	1136198	1.15	43.39
5718016	181000	3.14	96.83	2118106	8.47	62.96	749207	4.22	86.90	2185049	3.19	61.78
เปอร์เซ็นต์การลดเฉลี่ย			**60.11**			**29.73**			**28.63**			**27.59**

4.4.9 ผลการทดลองการลดขนาดข้อมูลชนิด **WMV** ด้วยวิธีต่างๆ

ตารางที่ 4.4.9 เป็นตารางแสดงผลการทดลองการลดขนาดไฟล์ภาพเคลื่อนไหวประเภท **WMV** ด้วยวิธีต่างๆ จากผลการทดลองที่ได้จะเห็นว่าประสิทธิภาพในการลดขนาดด้วยวิธีต่างๆให้ผลลัพธ์ที่ไม่ดีนัก เพราะไฟล์ประเภท **WMV** ได้ผ่านการบีบอัดมาก่อนแล้วนั่นเอง โดยวิธี **WINZIP** มีประสิทธิภาพในการลดขนาดอยู่ที่ 7.58% วิธี **Adaptive Huffman** มีประสิทธิภาพเฉลี่ยที่ 0.64% ส่วนวิธี **LZW** และ **Huffman** เมื่อทำการลดขนาดจะได้ไฟล์ที่มีขนาดเพิ่มขึ้น ซึ่งหมายความว่าไม่เหมาะสมที่จะใช้ทั้ง 2 วิธีนี้ในการลดขนาดไฟล์ประเภท **WMV**

ตารางที่4.4.9 แสดงผลการลดขนาดไฟล์ WMV ด้วยวิธีต่างๆ

ขนาดไฟล์เริ่มต้น (Bytes)	WINZIP			ADAPTIVE HUFFMAN			LZW			HUFFMAN		
	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด	ขนาด (Bytes)	เวลา (sec)	%การลด
1907023	1660270	4.21	12.94	1892570	4.02	0.76	1904243	3.89	0.15	2145698	3.54	-12.51
818893	764896	3.14	6.59	812491	3.18	0.78	819633	3.08	-0.09	924843	3.02	-12.93
2215931	2023918	4.51	8.67	2206655	4.53	0.42	2228738	4.61	-0.57	2503826	4.23	-12.99
2537113	2433211	4.87	4.09	2528632	4.69	0.33	2551548	4.69	-0.56	2869555	4.31	-13.10
3001157	2832652	5.03	5.61	2974060	4.92	0.90	2997311	4.94	0.13	3379505	4.85	-12.60
เปอร์เซ็นต์การลดเฉลี่ย			**7.58**			**0.64**			**-0.18**			**-12.82**

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

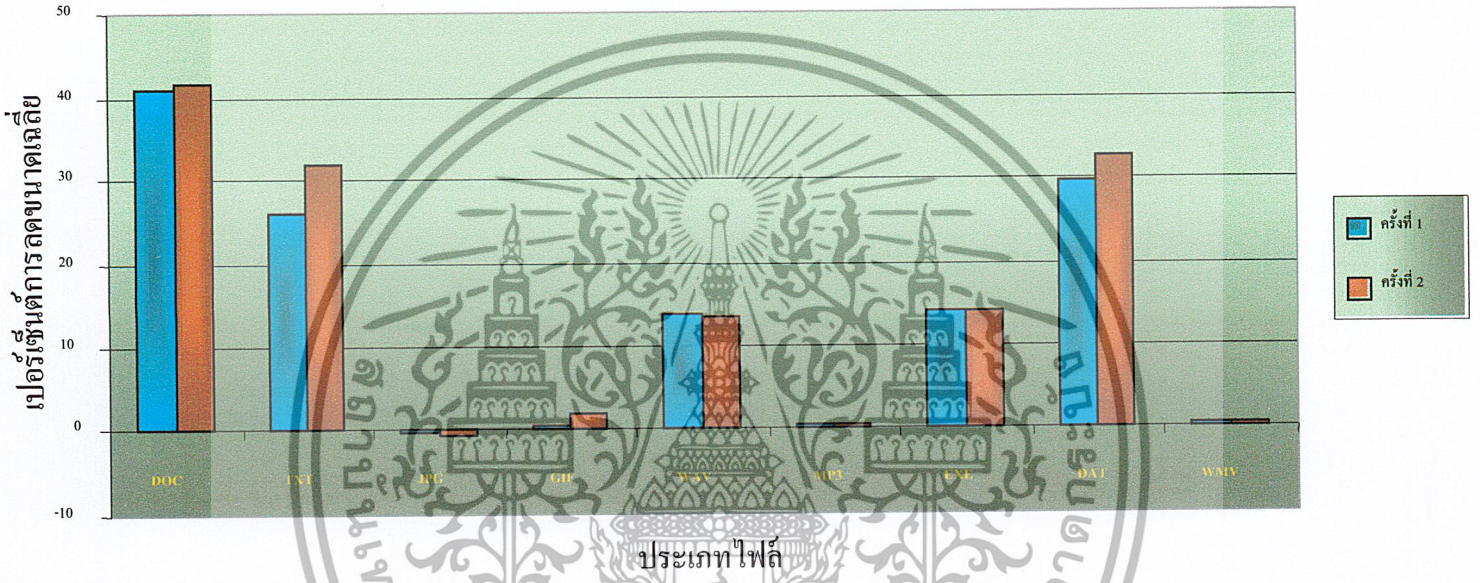
5.1 สรุปผลการทดลอง

จากตารางสรุปผลข้างต้นนั้น เราพอที่จะสามารถที่จะสรุปได้ว่า การลดขนาดด้วยวิธี **Adaptive Huffman** นั้นเหมาะสมที่จะใช้ลดขนาดไฟล์ประเภท **DOC** และ **TXT** หรือก็คือไฟล์ประเภทตัวอักษรนั่นเอง ส่วนไฟล์ประเภทที่ไม่เหมาะสมที่จะใช้วิธี **Adaptive Huffman** ในการลดขนาดคือ ไฟล์จำพวกรูปภาพ กราฟฟิก หรือไฟล์ประเภทที่ถูกบีบลดขนาดมาก่อนแล้ว เพราะหลังจากการลดขนาดเราจะได้ไฟล์ใหม่ที่มีขนาดเกือบจะเท่ากับไฟล์ตัวเดิมหรืออาจจะมากกว่าด้วยซ้ำ

ตารางที่ 5.1 ตารางแสดงผลการลดขนาดไฟล์เฉลี่ยประเภทต่างๆด้วยวิธี Adaptive Huffman

ประเภทไฟล์	เปอร์เซ็นต์การลดขนาดเฉลี่ยด้วยวิธี Adaptive Huffman	
	ครั้งที่ 1	ครั้งที่ 2
DOC	41.10	41.52
TXT	26.18	31.98
JPG	-0.34	-0.52
GIF	0.42	1.84
WAV	13.87	13.52
MP3	0.48	0.47
EXE	14.29	14.39
DAT	29.73	32.48
WMV	0.64	0.62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 ผลการเปรียบเทียบผลการลดขนาดไฟล์ต่างๆด้วยวิธี Adaptive Huffman

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

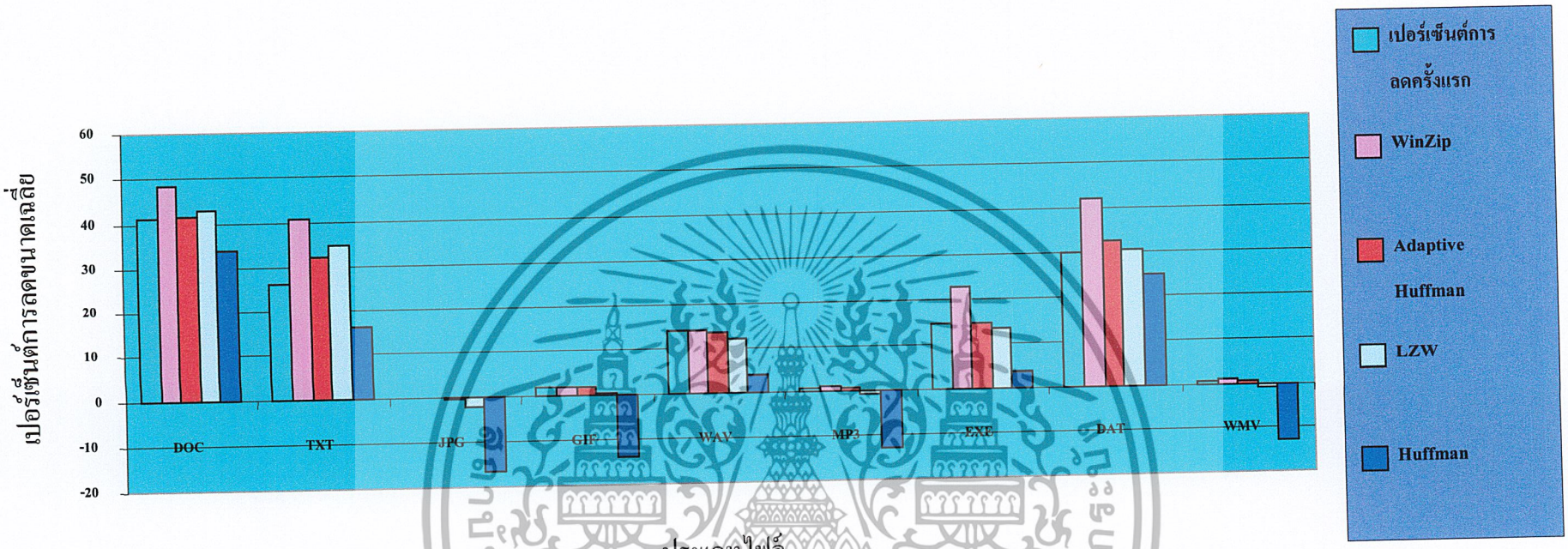
5.2 สรุปผลการลดขนาดครั้งที่ 2 หลังจากลดขนาดครั้งแรกด้วย Adaptive Huffman

จากผลการทดลองที่ได้เราจะเห็นได้ว่าในไฟล์ประเภท **DOC TXT EXE** และ **DAT** หลังจากทำการลดขนาดครั้งแรกไปแล้วยังสามารถทำการลดขนาดลงได้อีกในครั้งที่ 2 โดยเฉพาะวิธี **WINZIP** ซึ่งสามารถลดขนาดลงมาได้อีกเกือบหนึ่งเท่าตัว ซึ่งสามารถแสดให้เห็นว่าการลดขนาดไฟล์ด้วยวิธี **Adaptive Huffman** ไม่สามารถทำการลดขนาดและแปลงไฟล์ให้มีความละเอียดเพียงพอที่จะทำการลดขนาดในครั้งเดียวได้

ตารางที่ 5.2 แสดงการเปรียบเทียบผลการลดขนาดครั้งที่ 2 หลังจากลดด้วย Adaptive Huffman

ประเภทไฟล์	เปอร์เซ็นต์การลดขนาดครั้งแรกด้วย Adaptive Huffman	WINZIP	Adaptive Huffman	LZW	Huffman
DOC	41.1	48.38	41.52	42.79	33.55
TXT	26.18	40.66	31.98	34.68	16.25
JPG	-0.19	-0.33	-0.52	-2.17	-16.38
GIF	2.13	1.98	1.84	0.21	-13.83
WAV	13.87	13.92	13.52	12.40	4.10
MP3	0.48	1.15	0.47	-1.07	-12.93
EXE	14.29	22.78	14.39	13.24	3.67
DAT	29.73	41.95	32.48	30.22	24.79
WMV	0.64	0.98	0.62	-0.90	-13.08

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แผนภูมิแสดงการเปรียบเทียบการลดขนาดครั้งที่ 2 หลังจากลดด้วยวิธี Adaptive Huffman

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

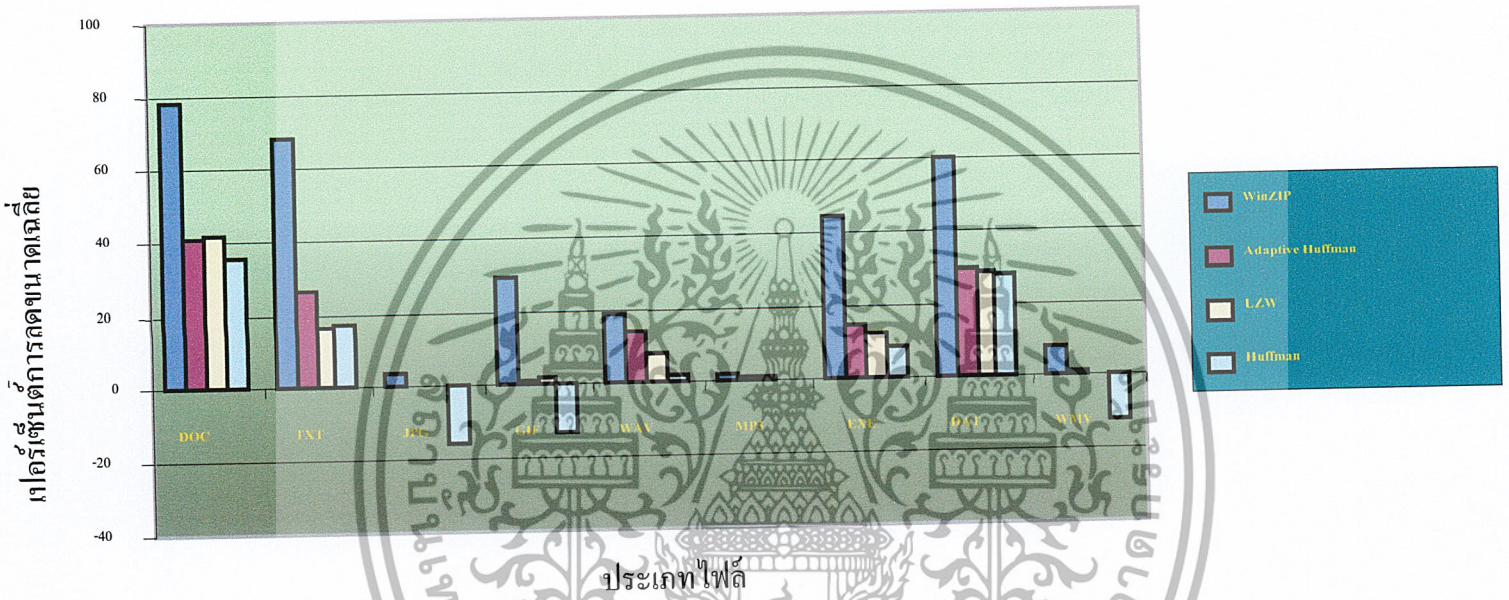
5.3 สรุปผลการทดลองเทียบกับวิธีอื่นๆ

จากตารางข้างต้นนี้จะเห็นได้ว่าวิธีที่ให้ประสิทธิภาพดีที่สุดในการลดขนาดก็คือ **WINZIP** นั่นเอง แต่ถ้าเราลองเปรียบเทียบแค่ 3 วิธีหลังจะเห็นว่า วิธี **Adaptive Huffman** จะให้ผลลัพธ์ที่มีประสิทธิภาพที่ดีที่สุด โดยถ้าเทียบกับวิธี **Huffman** จะเห็นว่าวิธี **Adaptive Huffman** มีประสิทธิภาพมากกว่าวิธี **Huffman** ถึงเกือบเท่าตัวเลยที่เดียว โดยในเฉพาะไฟล์ที่ไม่ใช่ข้อมูลประเภทตัวอักษรจะมีประสิทธิภาพต่างกันอย่างชัดเจน จากข้อมูลที่มีทำให้เราพอที่จะสามารถสรุปได้ว่า วิธี **Adaptive Huffman** มีประสิทธิภาพที่ดีกว่าวิธี **LZW** และ **Huffman** หรือหมายความว่าวิธี **Adaptive Huffman** น่าใช้มากกว่านั่นเอง

ตารางที่ 5.3 แสดงผลการทดลองลดขนาดข้อมูลเฉลี่ยเปรียบเทียบกันด้วยวิธีต่างๆ

ประเภทไฟล์	WINZIP	Adaptive Huffman	LZW	Huffman
DOC	78.83	41.10	42.08	35.30
TXT	68.46	26.18	16.59	17.30
JPG	2.80	-0.34	-0.65	-15.92
GIF	29.43	0.42	1.06	-13.39
WAV	18.96	13.87	7.31	1.69
MP3	1.76	0.48	0.23	-
EXE	44.15	14.29	11.80	7.82
DAT	60.11	29.73	28.63	27.59
WMV	7.58	0.64	-0.18	-12.82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แผนภูมิเปรียบเทียบประสิทธิภาพในการลดขนาดไฟล์ประเภทต่างๆ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

5.4 ข้อจำกัดการใช้โปรแกรม

5.4.1 โปรแกรม **Adaptive Huffman** ต้องทำการ RUN บนระบบ DOS เท่านั้น

5.4.2 จะต้องทำการระบุ path ไว้หน้า FILE.in ในกรณีที่ต้องการลดขนาดไฟล์ที่อยู่ต่างโฟลเดอร์กับโปรแกรม

5.4.3 จะต้องทำการระบุ path ไว้หน้า FILE.out ในกรณีที่ต้องการให้ไฟล์ที่ลดขนาดเรียบร้อยแล้วไปอยู่ในโฟลเดอร์ที่ต้องการ

5.5 ข้อเสนอแนะ

จากผลการทดลองที่ได้ จะเห็นว่าประสิทธิภาพในการลดขนาดไฟล์ประเภท **JPG GIF** และ **MP3** มีน้อยมาก โดยเฉพาะไฟล์ **JPG** ทั้งนี้เป็นเพราะไฟล์ **JPG** เป็นไฟล์กราฟฟิกที่ได้ถูกบีบอัดมาก่อนแล้ว โดยที่ข้อมูลในไฟล์ **JPG** จะถูกแบ่งย่อยออกมาและจัดเก็บในรูปเมทริกซ์จัตุรัส จึงทำให้ข้อมูลมีโอกาสซ้ำกันน้อยมาก ดังนั้นเมื่อนำมาทำการบีบอัดอีกครั้งด้วยวิธี **Adaptive Huffman** ผลลัพธ์ที่ได้จึงมีประสิทธิภาพที่ไม่น่าพอใจ

เมื่อนำข้อมูลที่ได้ทำการลดขนาดแล้วมาลดขนาดอีกครั้งด้วยวิธี **Adaptive Huffman** จะพบว่าไฟล์บางชนิดสามารถลดขนาดได้มากกว่าเดิมและเมื่อนำข้อมูลที่ถูกลดขนาดด้วยโปรแกรม **Winzip** แล้วมาทำการลดอีกครั้งโดยใช้วิธี **Adaptive Huffman** จะพบว่าไม่สามารถลดขนาดของข้อมูลได้มากกว่าเดิมแสดงให้เห็นว่าวิธีการ **Adaptive Huffman** ไม่สามารถแปลงข้อมูลเป็นโค้ดที่ละเอียดพอทำให้ยังเกิดการซ้ำของข้อมูลซึ่งทำให้ได้ผลตามที่กล่าวมาจึงเป็นที่น่าสนใจที่จะศึกษาหาวิธีการใหม่ๆ เพื่อให้การลดขนาดมีประสิทธิภาพมากขึ้นต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ภัคคินี ชิตสกุล. 2543. “เอกสารประกอบการเรียนวิชา Discrete Mathematics.” สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วิทยา เรืองพรวิสุทธิ์. **คู่มือโปรแกรมภาษาซี**. กรุงเทพฯ : บริษัท เอช.เอ็ม.กรุ๊ป จำกัด

Ellis Horowitz and Sartaj Sahni. 1990. **Data Structures**.Third Edition.

Gilbert Held and Thomas R. Marshell. **Data Compression : Techniques and Application Hardware and Software Considerations**. Second Edition.

Mark Nelson and Gailly J. 1996. **The Data Compression Book**. Second Edition.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

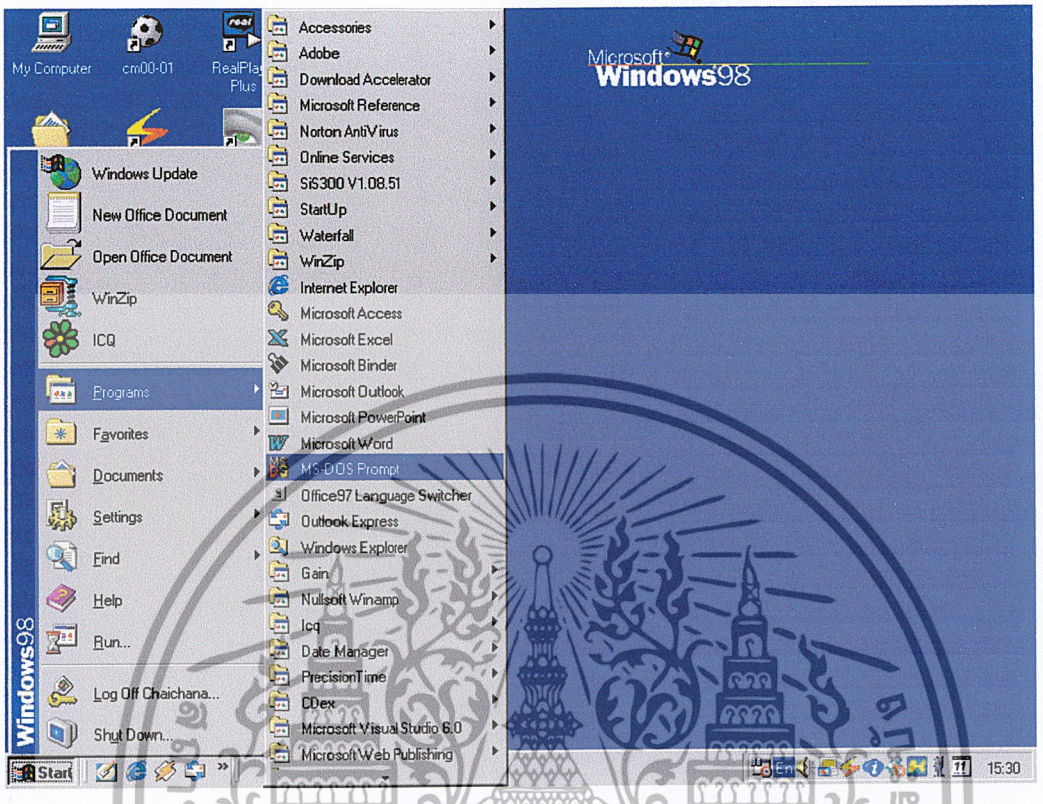
ภาคผนวก ก

ตัวอย่างการใช้โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้งานโปรแกรม

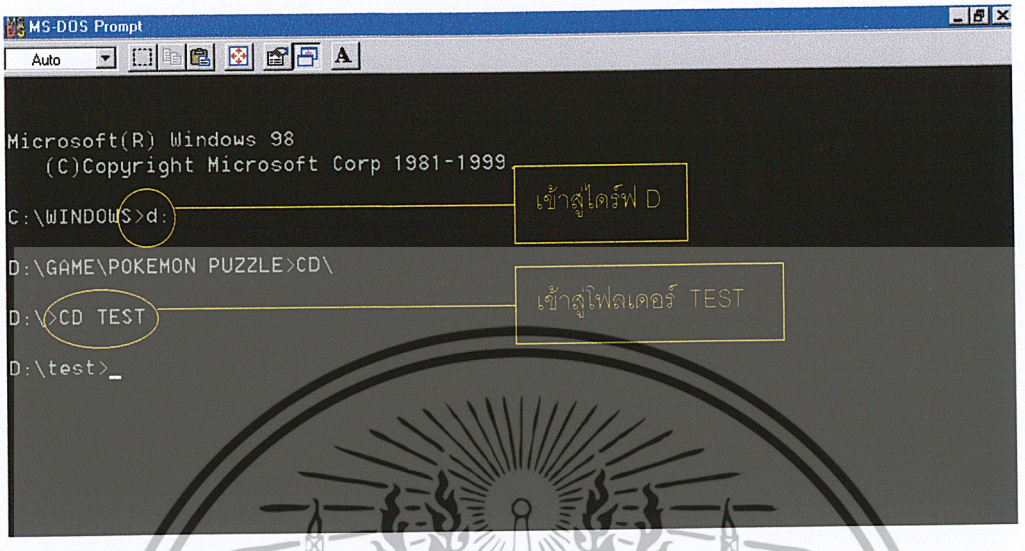
1. ทำการเข้าสู่ระบบ **DOS** โดยไปที่เมนู Start->Programs->MS-DOS prompt ดังรูป



รูปที่ ก.1 รูปภาพแสดงการเข้าสู่ระบบ DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

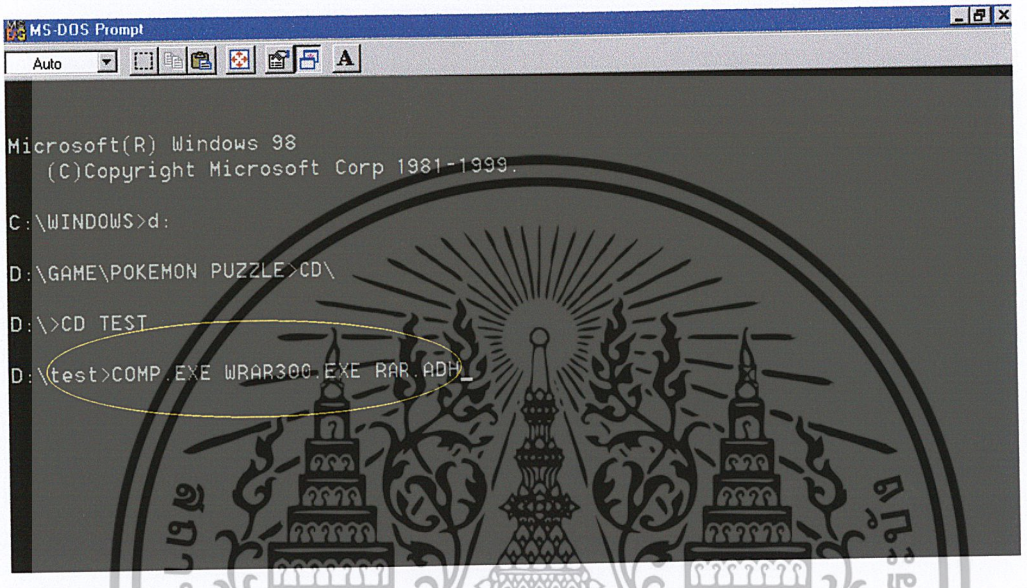
2. เมื่อเข้าสู่หน้าจอ **DOS** แล้วให้เราเข้าไปในไดเร็กทอรีที่เก็บโปรแกรม **Adaptive Huffman** ไว้ ตัวอย่างเช่นโปรแกรมถูกเก็บอยู่ในไดร์ฟ D อยู่ในโฟลเดอร์ที่ชื่อว่า **TEST** เราก็จะต้องเข้าไปที่ไดร์ฟ D แล้วพิมพ์คำว่า **CD TEST** เพื่อเข้าไปในโฟลเดอร์ **TEST** ดังรูป



รูปที่ ก.2 รูปภาพแสดงวิธีเข้าสู่โฟลเดอร์ที่ต้องการในระบบ DOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

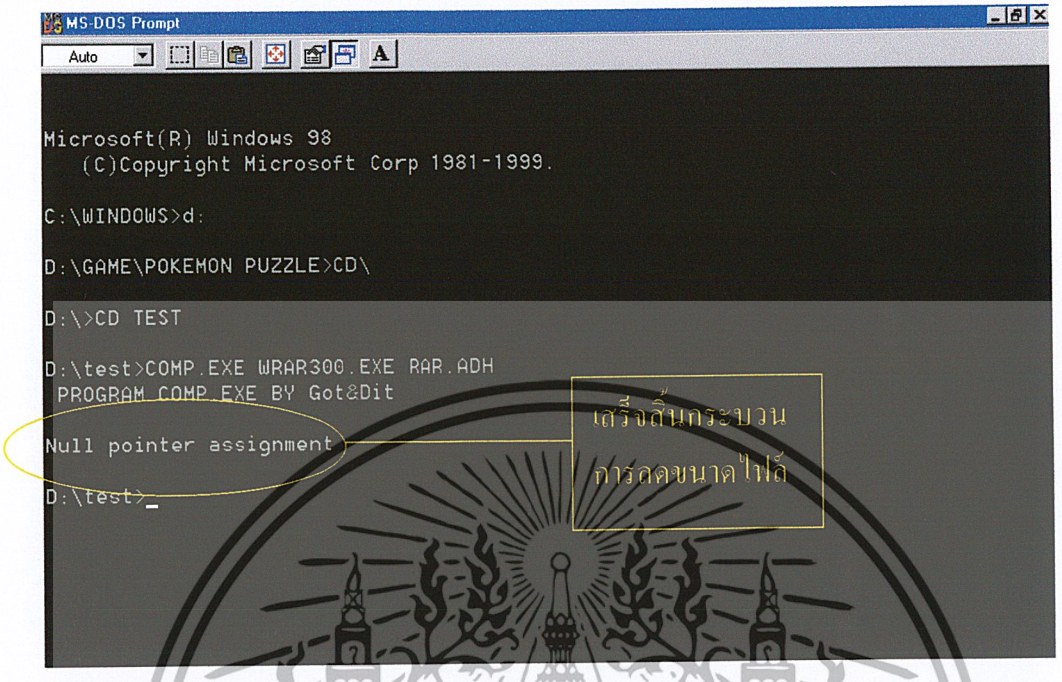
3. ทำการเรียกใช้โปรแกรมโดยพิมพ์คำว่า **COMP.EXE** แล้วเว้นวรรคหนึ่งทีตามด้วยชื่อและนามสกุลของไฟล์ที่ต้องการลดขนาด แล้วเว้นวรรคอีกหนึ่งทีตามด้วยชื่อไฟล์ที่ต้องการหลังการลดขนาด ตัวอย่างเช่น ต้องการย่อขนาดไฟล์สกุล **EXE** ชื่อ **WRAR300** (ถ้าไฟล์ที่ต้องการอยู่ต่างโฟลเดอร์กันต้องทำการระบุ path และชื่อโฟลเดอร์นั้นหน้า **FILE.in** ด้วย) และต้องการให้ไฟล์หลังทำการลดขนาดชื่อ **RAR.ADH** เราจะต้องพิมพ์ว่า **COMP.EXE WRAR300.EXE RAR.ADH** แล้วกด **ENTER** เพื่อทำการ **RUN** โปรแกรม



รูปที่ ก.3 รูปภาพแสดงวิธีการใช้โปรแกรม Adaptive Huffman

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

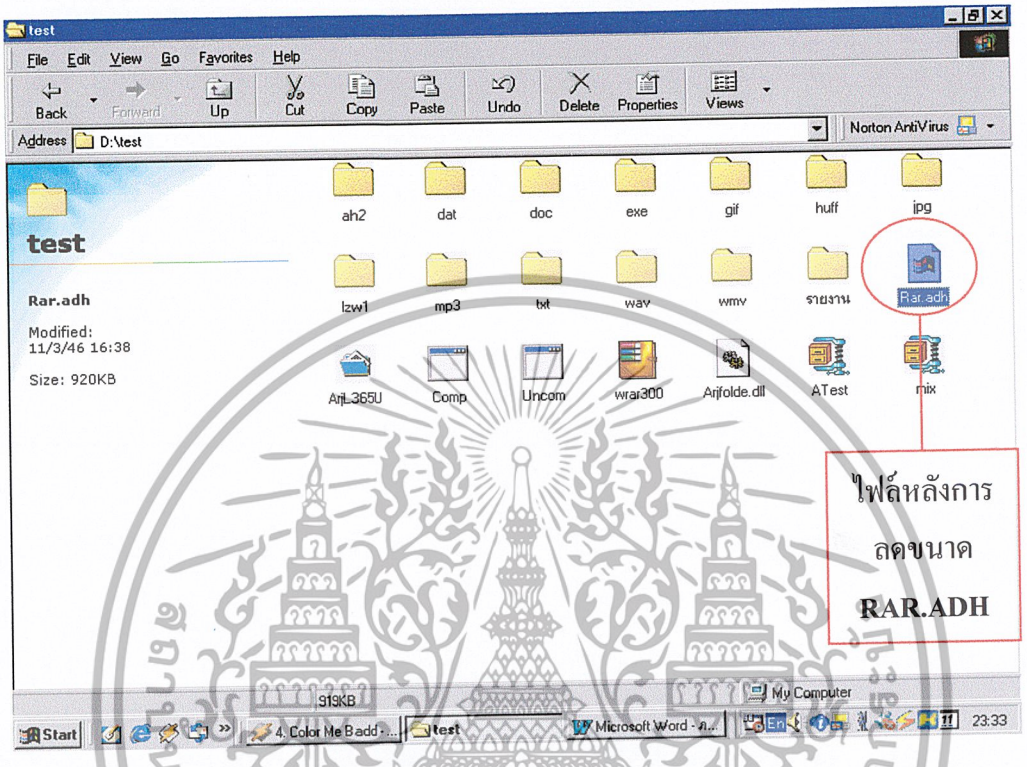
4. เมื่อทำการลดขนาดไฟล์เรียบร้อยแล้วจะแสดงคำว่า Null pointer assignment แสดงว่าเสร็จสิ้นการ **RUN** โปรแกรมแล้วนั่นเอง



รูปที่ ก.4 รูปภาพแสดงเวลาโปรแกรมทำการลดขนาดเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ไฟล์ที่ได้หลังจากการลดขนาดจะไปอยู่ที่โฟลเดอร์เดียวกับโฟลเดอร์ที่เก็บโปรแกรม **Adaptive Huffman** นั่นคือโฟลเดอร์ **TEST** นั่นเอง (ถ้าต้องการให้ไปอยู่โฟลเดอร์อื่นให้ทำการระบุ PATH และชื่อโฟลเดอร์ที่ต้องการไว้หน้า **FILE.out**) โดยไฟล์ที่ได้จะมีชื่อ **RAR** และมีนามสกุล **ADH** ดังรูป



รูปที่ ก.5 รูปภาพแสดงไฟล์หลังจากการลดขนาดเรียบร้อยแล้ว

ในส่วนของการใช้โปรแกรมขยายขนาดไฟล์นั้นก็จะใช้งานในลักษณะเดียวกับโปรแกรมบีบอัด เพียงแต่เราพิมพ์ **UNCOM.EXE** แทน **COMP.EXE** ในขั้นตอนการเรียกใช้โปรแกรมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

อธิบายการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการใช้โปรแกรม

1. โปรแกรม Comp.c

โปรแกรม **Comp.c** ประกอบด้วยฟังก์ชันย่อยๆ ซึ่งสอดคล้องกับวิธีการที่กล่าวมาแล้วในหัวข้อ **2.5.2 (บทที่ 2)** ดังนี้

1.1 Usage()

แนะนำผู้ใช้เกี่ยวกับการเรียกใช้โปรแกรม(กรณีที่เรียกใช้ไม่ถูกต้อง)

1.2 Transmit()

ทำหน้าที่ส่งตัวอักขระ ไปยัง Tree โดยมีฟังก์ชันย่อยดังนี้

1.2.1 Send()

ทำหน้าที่กำหนดบิต 0 และบิต 1 ให้กับ Tree โดยตัวลูกที่อยู่ทางขวา จะกำหนดให้เป็นบิต 1 และตัวลูกที่อยู่ทางซ้ายจะกำหนดให้เป็นบิต 0

1.2.2 Add_bit()

ทำหน้าที่ใส่บิตลงในโหนดที่ถูกกำหนดโดยฟังก์ชัน Send()

1.3 Add_ref()

ทำหน้าที่ Update ข้อมูล Tree โดยมีฟังก์ชันย่อยดังนี้

1.3.1 Increment()

ที่ทำหน้าที่เพิ่มค่าน้ำหนักให้แก่โหนดที่ได้รับตัวอักษรเพิ่มเข้าไป

1.3.2 Swap()

ทำหน้าที่สลับเลขตำแหน่งของโหนดที่มีการเปลี่ยนแปลง

1.3.3 Swaplist()

ทำหน้าที่สลับโหนดตามวิธีการของ Adaptive Compression (ดูรายละเอียดเพิ่มเติมใน 1.3.3.1)

1.3.4 Scott()

ทำหน้าที่เชคค่าน้ำหนัก (ดูรายละเอียดเพิ่มเติมใน 1.3.4.1)

1.3.3.1 Swaplist()

Swaplist() ทำหน้าที่สลับโหนดโดยทำงานดังนี้ถ้าโหนดด้านซ้ายมือของโหนดที่มีค่าน้ำหนักมากกว่า มีค่าน้ำหนักน้อยกว่าค่าน้ำนักของโหนดตัวอักษรตัวนั้นให้ทำการสลับที่โหนดตัวอักษรกับโหนดตัวนั้น(โดยย้ายโหนดที่เป็น **Descendent** ตามไปด้วย)

1.3.4.1 Scott()

Scott() เป็นฟังก์ชันที่ทำหน้าที่เชคค่าน้ำหนักของโหนดนำค่าน้ำหนักของของโหนดตัวอักษรนั้นไปเทียบกับค่าน้ำหนักของโหนดอื่นจนกว่าจะพบโหนดที่มีค่าน้ำหนักมากกว่าตัวแรกหรือจนกว่าจะพบโหนดราก (Root Node) โดยไล่จากโหนดที่อยู่ทางขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของโหนดตัวอักษรนั้นไปทางขวาเรื่อย ๆ ถ้าหาจนถึงตัวขวาสุดของชั้น (Level) นั้นแล้วยังไม่พบให้ไปหาต่อที่ชั้นถัดไปด้านบนโดยไล่จากซ้ายไปขวา

2. โปรแกรม Uncom.c

โปรแกรม Uncom.c ประกอบด้วยฟังก์ชันต่างๆ ดังนี้

2.1 Usage()

แนะนำการใช้งานเช่นเดียวกับโปรแกรม Comp.c

2.2 Receive()

รับตัวอักขระไปยัง Tree โดยมีฟังก์ชันย่อยดังนี้

2.2.1 Get_bit();

รับข้อมูลทีละบิตจากไฟล์ Input ที่เก็บอยู่ในหน่วยความจำ Buffered

2.3 Add_ref()

ทำการ Update Tree เช่นเดียวกับโปรแกรม Comp.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้