สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเตอร์เน็ต

DATA SORTING SIMULATION ON INTERNET

T039674

เจษฎา ยอดยิ่ง
วรรณา เชาวน์ชูวงศ์
อิทธิพล โตสงวน

ปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เลขหมู่.....
เลขทะเบียน 39674
วัน, เดือน, ปี 19 มิ.ย. 2544
.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SORTING SIMULATION ON INTERNET

JEDSADA YODYING

WANNA CHAOCHUOWONG

ITTIPOL TOSANGUAN

A SPECIAL PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCES

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LARDKRABANG

ACADEMIC YEAR 2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาพิเศษเรื่อง	โปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเตอร์เน็ต	
ชื่อนักศึกษา	นายเจษฎา ยอดยิ่ง	40056016
	นางสาววรรณา เชาวน์ชูวงศ์	40056071
	นายอิทธิพล โตสงวน	40056122
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	ดร.จันทร์บูรณ์ สถิตวิริยวงศ์	

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปัญหาพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ประจำปีการศึกษา 2543

คณะกรรมการสอบ		ลายมือชื่อ
ประธานกรรมการ	ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์	
กรรมการ	อาจารย์วีระชัย ตันยะสิทธิ์	
กรรมการและอาจารย์ที่ปรึกษา	ดร.จันทร์บูรณ์ สถิตวิริยวงศ์	

(ผู้ช่วยศาสตราจารย์ไพรบูลย์ พันธรักษ์พงษ์)
หัวหน้าภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

ลิขสิทธิ์ของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปัญหาพิเศษ	โปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเทอร์เน็ต	
ชื่อนักศึกษา	นายเจษฎา ยอดยิ่ง	40056016
	นางสาววรรณา เชาวน์ชูวงศ์	40056071
	นายอิทธิพล โตสงวน	40056122
ปริญญา	วิทยาศาสตรบัณฑิต	
ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์	
สาขาวิชา	วิทยาการคอมพิวเตอร์	
ปีการศึกษา	2543	
อาจารย์ที่ปรึกษา	ดร. จันทร์บูรณ์ สถิตวิริยวงศ์	

บทคัดย่อ

ในปัจจุบันเป็นที่ยอมรับกันอย่างกว้างในสถาบันการศึกษาว่า โปรแกรมช่วยสอนมีความสำคัญและมีความจำเป็นต่อการศึกษาเพื่อเพิ่มประสิทธิภาพของการศึกษา ซึ่งการศึกษาจะมีประสิทธิภาพและสำเร็จได้จะต้องมีสื่อและอุปกรณ์ทางการศึกษาที่ทันสมัยและหนึ่งในทางเลือกนั้นคือสื่อการเรียนการสอนผ่านทางเครือข่าย อินเทอร์เน็ต

ดังนั้นโปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเตอร์เน็ตจึงได้ถูกพัฒนาขึ้นเพื่อเพิ่มประสิทธิภาพการเรียนการสอนโดยนักเรียน นักศึกษา หรือผู้ที่สนใจสามารถใช้งานโปรแกรมจำลองการจัดเรียงข้อมูลผ่านทางเครือข่ายอินเตอร์เน็ตได้ โดยโปรแกรมจำลองการจัดเรียงข้อมูลนี้ถูกพัฒนาขึ้นด้วยโปรแกรม Macromedia Flash 5.0 เป็นหลัก โดยมีวัตถุประสงค์เพื่อให้ผู้ใช้เข้าใจรายละเอียดในแต่ละขั้นตอนของการจัดเรียงข้อมูลด้วย อัลกอริทึมต่างๆอีกทั้งได้เล็งเห็นถึงข้อแตกต่างในการจัดเรียงข้อมูลแต่ละวิธีเพื่อความสามารถในการเลือกใช้วิธีการจัดเรียงลำดับข้อมูลให้เหมาะสมกับลักษณะของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Data Sorting Simulation On Internet	
Students	MisterJedsada Yodying	40056016
	MissWanna Chaochuowong	40056071
	MisterIttipol Tosanguan	40056122
Degree	Bachelor's Degree of Science	
Department	Mathematics and Computer Sciences, Faculty of Science	
Programme	Computer Sciences	
Year	2000	
Special Project Advisor	Dr. Chanboon Saphitwiriyawong	

ABSTRACT

The widely used of an Internet in this modern society is considered as the way to enclose with the student who want to learn more by themselves. Therefore, this data sorting simulation program is developed for widely used by an Internet users.

This program aims to give the idea of each data sorting theme and visualize them by simulating its algorithm. It also provides an effective and well- organized exercise in which divided into two parts (Introduction Exercise and Practice Exercise). The exercise is subject to give a chance for practicing and introducing the users into the program.

To sum up, with a good and up to date resources, this program is suitable for all users who want toget an clear idea of each data sorting algorithm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปัญหาพิเศษเรื่องโปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเทอร์เน็ต สามารถสำเร็จลุล่วงไปด้วยดี คณะผู้จัดทำต้องขอขอบพระคุณ ดร. จันทร์บูรณ์ สถิตวิริยวงศ์ อาจารย์ที่ปรึกษาผู้รับผิดชอบปัญหาพิเศษฉบับนี้ที่กรุณาให้คำแนะนำและเป็นที่ปรึกษาอันก่อให้เกิดแนวความคิดในการแก้ปัญหาต่างๆในการจัดทำปัญหาพิเศษฉบับนี้รวมทั้งเป็นผู้ตรวจสอบความถูกต้องและแนะแนวทางในการดำเนินงานด้วยความเอาใจใส่เป็นอย่างยิ่ง

นอกจากนี้คณะผู้จัดทำต้องขอขอบพระคุณบิดาและมารดาที่ให้ความสนับสนุนทั้งทางด้านกำลังใจและทุนทรัพย์จนการทำปัญหาพิเศษครั้งนี้สำเร็จด้วยดี และต้องขอขอบคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ที่ให้ความสะดวกในการใช้ห้องปฏิบัติการคอมพิวเตอร์และให้ความสะดวกในการเบิกอุปกรณ์ต่างๆ ที่ใช้ในการทำปัญหาพิเศษ รวมทั้งเพื่อนๆ และพี่ๆ ทุกคนที่ให้ความช่วยเหลือในด้านต่างๆ เป็นอย่างดีเกี่ยวกับการทำปัญหาพิเศษไว้ ณ ที่นี้

คณะผู้จัดทำ

มีนาคม 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....V

สารบัญภาพ.....VI

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 สมมุติฐานของการศึกษา.....1

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา.....1

1.5 ขอบเขตการศึกษา.....2

1.6 ขั้นตอนการศึกษา.....2

บทที่ 2 ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 การเขียนโปรแกรมบนเว็บ (Web Programming).....3

2.1.1 อินเตอร์เน็ตคืออะไร.....3

2.1.2 World Wide Web (WWW) คืออะไร.....3

2.1.3 ส่วนประกอบของ WWW.....3

2.1.4 IP Address, DNS, E-mail Address และ URL คืออะไร.....4

2.2 แบบจำลองคืออะไร.....6

2.3 การหาฟังก์ชัน Big-Oh.....7

2.4 การเรียงลำดับ (Sorting).....9

2.4.1 Priority Queue Sorting Methods.....10

2.4.1.1 Selection Sort.....10

2.4.1.2 Heap Sort.....12

2.4.2 Divide-and-Conquer Methods.....19

2.4.2.1 Quick Sort.....20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4.2.2 Merge Sort 24
2.4.3 Transposition Sorting 25
2.4.3.1 Bubble Sort 25
2.4.4 Diminishing Increment Sorting 27
2.4.4.1 Shell Sort 27
2.4.5 Address Calculation Sorting 28
2.4.5.1 Proxmap Sort 28
2.4.5.2 Radix Sort 29
2.4.6 Inset and keep Sorted 29
2.4.6.1 Insertion Sort 29
2.4.6.2 Tree Sort 35
2.4.7 Comparison and Perspective.....37
2.5 Active Server Pages 38
2.5.1 Active Server Pages คืออะไร 38
2.5.2 Active Server Pages ทำอะไรได้บ้าง 39
2.5.3 Active Server Pages ทำงานอย่างไร 40
2.5.4 ความต้องการของระบบในการใช้งาน ASP.....40
2.5.5 รูปแบบการเชื่อมต่อฐานข้อมูล.....41
2.5.5.1 การติดต่อฐานข้อมูลโดยใช้ ODBC.....41
2.5.5.2 การติดต่อฐานข้อมูลโดยใช้ DSNLess.....44

บทที่ 3 วิธีดำเนินการวิจัย

3.1 ขั้นตอนในการดำเนินงาน.....45
3.2 คุณลักษณะของโปรแกรมที่จะออกแบบและพัฒนา.....45
3.3 การกำหนดเนื้อหา.....46
3.4 โปรแกรมประยุกต์ที่ใช้ในการพัฒนาโปรแกรม.....46
3.5 อุปกรณ์ที่ใช้ในการพัฒนาโปรแกรม.....47
3.6 แผนภาพการทำงานของระบบ.....47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.6.1 โครงสร้างการทำงานของระบบโดยรวม.....47
3.6.2 โครงสร้างการทำงานของหน้าจอเนื้อหาการจัดเรียงข้อมูลแต่ละประเภท..48
3.6.3 โครงสร้างการทำงานของหน้าจอระบบแบบฝึกหัด.....51
3.6.4 แผนภาพแสดง Data Flow ของระบบ.....53
3.6.5 Domain ของระบบแบบฝึกหัด.....55
3.6.6 รายละเอียดเกี่ยวกับฐานข้อมูลที่ใช้ในการพัฒนาระบบ.....56
3.6.6.1 ฐานข้อมูล Sortdb.....56
3.6.6.2 ฐานข้อมูล Registeration.....58
3.6.6.3 ฐานข้อมูล Temp.....59
3.6.7 แผนภาพ ER Diagram ของระบบแบบฝึกหัด.....60

บทที่ 4 การพัฒนาระบบ

4.1 หลักการทำงานของโปรแกรมสาธิตการจัดเรียงข้อมูล.....61
4.1.1 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort.....61
4.1.2 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort.....69
4.1.3 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Merge Sort.....76
4.1.4 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Radix Sort.....86
4.1.5 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort.....88
4.1.6 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Heap Sort.....91
4.1.7 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Shell Sort.....100
4.1.8 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Proxmap Sort.....103
4.1.9 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Tree Sort.....107
4.1.10 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Quick Sort.....109
4.2 หลักการทำงานของโปรแกรมจำลองการจัดเรียงข้อมูล.....119
4.2.1 ส่วนประกอบของหน้าจอการทำงานของโปรแกรม.....119
4.2.2 ตัวอย่างการทำงานของโปรแกรมการจัดเรียงข้อมูล.....77
4.3 การทำงานของระบบแบบฝึกหัด.....77
4.3.1 แสดงผลการทำงานของหน้าจอหลักๆของระบบแบบฝึกหัด.....82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 การพัฒนาระบบ

5.1 สรุปผลในภาพรวม.....128

5.2 สรุปผลส่วนของโปรแกรมสาธิตการจัดเรียงข้อมูล.....128

5.3 สรุปผลส่วนของโปรแกรมจำลองการจัดเรียงข้อมูล.....128

5.4 สรุปผลในส่วนของโปรแกรมระบบแบบฝึกหัด.....129

5.5 ข้อเสนอแนะ.....129

บรรณานุกรม.....130

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางผลลัพธ์ของตัวอย่างที่ 1 ในการหาฟังก์ชัน Big-O.....7

2.2 ตารางผลลัพธ์ของตัวอย่างที่ 2 ในการหาฟังก์ชัน Big-O.....8

2.3 ตารางผลลัพธ์ของตัวอย่างที่ 3 ในการหาฟังก์ชัน Big-O.....9

2.4 ตารางผลลัพธ์ของตัวอย่างที่ 4 ในการหาฟังก์ชัน Big-O.....9

2.5 ตารางแสดงเวลาที่ใช้ในการจัดเรียงข้อมูลด้วยวิธีต่างๆ.....37

3.1 ตารางแสดงรหัสประจำแต่ละ Sorting Theme (Sort).....56

3.2 ตารางแสดงรายการ Introduction Exercise.....56

3.3 ตารางแบบฝึกหัด.....57

3.4 ตารางรายการคำตอบในแต่ละตัวเลือกของโจทย์แต่ละข้อ.....57

3.5 ตารางบันทึกการทำแบบฝึกหัดของผู้ใช้โปรแกรม.....57

3.6 ตารางบันทึกคะแนนชั่วคราว.....58

3.7 ตารางรายการประวัติสมาชิก.....58

3.8 ตารางบันทึกประวัติผู้ควบคุมระบบ.....59

3.9 ตารางบันทึกสถานะผู้เข้าใช้ระบบอยู่ขณะนั้น.....81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 แสดงแผนภาพ Sorting Themes.....10
2.2 แสดงโปรซีเจอร์ Selection Sort.....11
2.3 แสดงขั้นตอนการใช้อัลกอริทึม Selection Sort กับข้อมูล 5 20 7 10 13 2 30.....12
2.4 แสดงต้นไม้ที่ได้จากชุดข้อมูล 12 10 4 6 29 15 30.....13
2.5 แสดงผลลัพธ์ที่ได้จากการที่ h_3 ในภาพที่ 2.4 ถูกแทนที่ด้วย h_7 และ h_7 ถูกเปลี่ยนค่าเป็น h_3.....13
2.6 แสดงผลลัพธ์ที่ได้จากการที่ h_2 ถูกแทนที่ด้วย h_5 และ h_5 ถูกเปลี่ยนค่าเป็น h_2.....14
2.7 แสดงการทำงานต่อเนื่องจากภาพที่ 2.6.....14
2.8 แสดงการนำต้นไม้ไม้ฮีปในภาพที่ 2.7 มาดำเนินการตามอัลกอริทึมฮีปซอร์ท.....14
2.9 แสดงการนำต้นไม้ไม้ฮีปในภาพที่ 2.7 มาดำเนินการตามอัลกอริทึมฮีปซอร์ท.....14
2.10 แสดงการนำต้นไม้ไม้ฮีปในภาพที่ 2.7 มาดำเนินการตามอัลกอริทึมฮีปซอร์ท.....15
2.11 แสดงการนำต้นไม้ไม้ฮีปในภาพที่ 2.7 มาดำเนินการตามอัลกอริทึมฮีปซอร์ท.....15
2.12 แสดงการนำต้นไม้ไม้ฮีปในภาพที่ 2.7 มาดำเนินการตามอัลกอริทึมฮีปซอร์ท.....15
2.13 แสดงการนำต้นไม้ไม้ฮีปในภาพที่ 2.7 มาดำเนินการตามอัลกอริทึมฮีปซอร์ท.....16
2.14 แสดง array h.....16
2.15 ต้นไม้ฮีป.....16
2.16 อาร์เรย์แทนต้นไม้ฮีปในภาพที่ 2.15.....16
2.17 โปรแกรม Heap Sort.....19
2.18 ตัวอย่างการจัดเรียงข้อมูลแบบ Ascending ด้วย Heap Sort.....21
2.19 โปรแกรม Quick Sort.....22
2.20 ตัวอย่างการจัดเรียงข้อมูลด้วย Quick Sort.....23
2.21 Algorithm ของ Merge Sort.....24
2.22 ตัวอย่างการทำงานด้วย Merge Sort.....25
2.23 Procedure Bubble Sort.....26
2.24 ตัวอย่างการทำงานด้วย Bubble Sort.....26
2.25 แสดงตัวอย่างการหาค่า Increment Sequence และการเรียงลำดับโดย Shell Sort.....27
2.26 ตัวอย่างการเรียงข้อมูลด้วย Radix Sort.....28
2.27 แสดงตัวอย่างการเรียงข้อมูลด้วยการแทรก.....30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
2.28 โปรซีเจอร์ Insertion Sort	33
2.29 แสดงตัวอย่างการเรียงข้อมูลด้วย Insertion Sort	33
2.30 แสดงตัวอย่างการเรียงข้อมูลด้วย Insertion Sort	34
2.31 ภาพต้นไม้ของข้อมูล 12 10 4 6 9 15 30 22	36
2.32 ภาพต้นไม้เมื่อดึงข้อมูล 4 ออกและแทนที่ด้วยค่าที่มากๆ	36
2.33 ภาพต้นไม้เมื่อทำการ Promote ค่า 6 ซึ่งมีค่าน้อยที่สุดมาไว้ที่ Root Node	36
2.34 หลักการทำงานของ ASP	40
2.35 แสดงหน้าจอเมื่อเลือกIcon ODBC บนหน้าจอ Control Panel	42
2.36 แสดงหน้าจอเมื่อเลือก Tab Data Source Name	42
2.37 แสดงหน้าจอรายการ Driver	43
2.38 แสดงหน้าจอODBC Microsoft Access 97	43
2.39 หน้าต่าง ODBC Data Source Administrator Dialogue	44
3.1 แสดงโครงร่างการทำงานของระบบโดยรวม	47
3.2 แสดงโครงร่างการทำงานของหน้าจอ Bubble Sort	48
3.3 แสดงโครงร่างการทำงานของหน้าจอ Insertion Sort	48
3.4 แสดงโครงร่างการทำงานของหน้าจอ Tree Sort	49
3.5 แสดงโครงร่างการทำงานของหน้าจอ Selection Sort	49
3.6 แสดงโครงร่างการทำงานของหน้าจอ Heap Sort	49
3.7 แสดงโครงร่างการทำงานของหน้าจอ Quick Sort	50
3.8 แสดงโครงร่างการทำงานของหน้าจอ Merge Sort	50
3.9 แสดงโครงร่างการทำงานของหน้าจอ Shell Sort	50
3.10 แสดงโครงร่างการทำงานของหน้าจอ Proxmap Sort	51
3.11 แสดงโครงร่างการทำงานของหน้าจอ Radix Sort	51
3.12 แสดงโครงร่างการทำงานในส่วนแบบฝึกหัดในส่วนของผู้ใช้โปรแกรม	51
3.13 แสดงโครงร่างการทำงานในส่วนแบบฝึกหัดในส่วนของAdministrator	52
3.14 แสดง DFD ระดับ 0 ของระบบโดยรวม	53
3.15 แสดง DFD ระดับ 1 ของ Process Do Exercise	54
3.16 แสดง DFD ระดับ 1 ของ Process Manage Exercises	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

3.17 แสดง ER Diagram ของระบบแบบฝึกหัด.....60
4.1 แสดงหน้าจอเริ่มต้นของโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort.....61
4.2 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 1 62
4.3 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 2.....63
4.4 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 3 63
4.5 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 4.....64
4.6 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 5.....64
4.7 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 6.....65
4.8 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 7.....65
4.9 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 8.....66
4.10 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 9.....66
4.11 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 10...67
4.12 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 11...67
4.13 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 12...68
4.14 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 36...69
4.15 แสดงหน้าจอเริ่มต้นของโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort.....70
4.16 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่1 ...70
4.17 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่2....71
4.18 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่3.....71
4.19 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่4....72
4.20 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 5...72
4.21 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 6...73
4.22 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 7...73
4.23 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 8...74
4.24 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 9...74
4.25 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 10.....75
4.26 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 11. 75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

4.27 แสดงหน้าจอเริ่มต้นของการสาธิตการ Merge ชุดข้อมูล....76

4.28 แสดงหน้าจอการสาธิตการ Merge ชุดข้อมูล(ต่อ)....77

4.29 แสดงหน้าจอการสาธิตการ Merge ชุดข้อมูล(ต่อ)....77

4.30 แสดงหน้าจอการสาธิตการ Merge ชุดข้อมูล(ต่อ)....78

4.31 แสดงหน้าจอการสาธิตการ Merge ชุดข้อมูล(ต่อ)....78

4.32 แสดงหน้าจอการสาธิตการ Merge ชุดข้อมูล(ต่อ)....79

4.33 แสดงหน้าจอการสาธิตการ Merge ชุดข้อมูล(ต่อ)....79

4.34 แสดงหน้าจอเริ่มต้นของการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort....80

4.35 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....81

4.36 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....81

4.37 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....82

4.38 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....82

4.39 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....83

4.40 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....84

4.41 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....84

4.42 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....85

4.43 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort (ต่อ)....85

4.44 แสดงหน้าจอเริ่มแรกของโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Radix Sort....86

4.45 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Radix Sort (ต่อ)....87

4.46 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Radix Sort (ต่อ)....87

4.47 แสดงหน้าจอสิ้นสุดของโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Radix Sort....88

4.48 แสดงหน้าจอเริ่มแรกของโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort....89

4.49 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Selection Sort (ต่อ)....89

4.50 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Selection Sort (ต่อ)....90

4.51 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Selection Sort (ต่อ)....90

4.52 แสดงหน้าจอสิ้นสุดของโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort....91

4.53 แสดงหน้าจอเริ่มแรกในการนำข้อมูล 12 10 4 6 29 15 30 มาสร้างเป็นต้นไม้ฮีป....92

4.54 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ)....92

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

4.55 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....93

4.56 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....93

4.57 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....94

4.58 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....94

4.59 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....95

4.60 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....95

4.61 แสดงหน้าจอในการ สร้างต้นไม้ฮีป(ต่อ).....96

4.62 หน้าจอแสดงต้นไม้ฮีปที่จะนำมาดำเนินการตามอัลกอริทึม Heap Sort.....96

4.63 แสดงหน้าจอในการดำเนินการตามอัลกอริทึม Heap Sort.....97

4.64 แสดงหน้าจอในการดำเนินการตามอัลกอริทึม Heap Sort (ต่อ).....97

4.65 แสดงหน้าจอในการดำเนินการตามอัลกอริทึม Heap Sort (ต่อ).....98

4.66 แสดงหน้าจอในการดำเนินการตามอัลกอริทึม Heap Sort (ต่อ).....98

4.67 แสดงหน้าจอสิ้นสุดการดำเนินการตามอัลกอริทึม Heap Sort 98

4.68 เริ่มต้นขบวนการแทนต้นไม้ด้วยอาร์เรย์.....99

4.69 ผลลัพธ์ของขบวนการแทนต้นไม้ด้วยอาร์เรย์.....100

4.70 แสดงหน้าจอการเรียงลำดับข้อมูลด้วย Shell Sort โดยเริ่มที่ค่า Delta = 5.....100

4.71 แสดงหน้าจอการเรียงลำดับข้อมูลในแต่ละช่วงจากนั้นจึงทำการเรียงข้อมูลใน Array A.....101

4.72 แสดงหน้าจอการเรียงลำดับข้อมูลด้วย Shell Sort โดยที่ค่า Delta = 3.....101

4.73 แสดงหน้าจอการเรียงลำดับข้อมูลในแต่ละช่วง.....102

4.74 แสดงหน้าจอการเรียงข้อมูลใน Array A.....102

4.75 แสดงหน้าจอสิ้นสุดการดำเนินการตามอัลกอริทึม Shell Sort 103

4.76 แสดงชุดข้อมูลเริ่มต้น A ที่จะทำการจัดเรียงโดย Proxmap Sort.....104

4.77 แสดงการหา Hit Counts จากสูตรที่กำหนด.....104

4.78 แสดงการหา Hit Counts จากสูตรที่กำหนด(ต่อ).....104

4.79 แสดงการหา Proxmap จากสูตรที่กำหนด.....105

4.80 แสดงการหา Insertion Locatio จากสูตรที่กำหนด.....105

4.81 แสดงการใส่ข้อมูลที่เรียงลำดับแล้วใน A_2.....106

4.82 แสดงผลลัพธ์ที่ได้จากการดำเนินการตามอัลกอริทึม Proxmap Sort 106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

4.83 แสดงการสร้าง Binary Tree จากชุดข้อมูล.....107

4.84 แสดง Binary Tree ที่ได้จากชุดข้อมูล.....108

4.85 แสดงการดึงค่าที่น้อยที่สุดออกจาก Root Node และแทนค่าด้วยตัวเลขที่มีค่ามากๆ.....108

4.86 แสดงหน้าจอสิ้นสุดการเรียงลำดับข้อมูลด้วย Tree Sort 109

4.87 แสดงหน้าจอเริ่มต้นการจัดเรียงข้อมูลด้วย Quick Sort 110

4.88 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 110

4.89 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 111

4.90 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 112

4.91 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 112

4.92 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 113

4.93 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 113

4.94 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 114

4.95 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 114

4.96 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 115

4.97 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 115

4.98 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 116

4.99 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 116

4.100 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 117

4.101 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 117

4.102 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 118

4.103 แสดงหน้าจอการจัดเรียงข้อมูลด้วย Quick Sort (ต่อ) 118

4.104 แสดงหน้าจอสิ้นสุดการจัดเรียงข้อมูลด้วย Quick Sort 119

4.105 แสดงหน้าจอเตรียมรับตัวเลขจากผู้ใช้.....120

4.106 แสดงหน้าจอเมื่อทำการจัดเรียงชุดข้อมูลที่มีข้อมูล 1 ตัว.....121

4.107 แสดงหน้าจอโปรแกรมจำลองการจัดเรียงข้อมูลเมื่อได้รับชุดข้อมูล.....121

4.108 แสดงหน้าจอ Login User.....122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

4.109 หน้าจอการลงทะเบียนผู้ใช้โปรแกรม.....123
4.110 หน้าจอเมื่อผู้ใช้ลืม Password และต้องให้ข้อมูลเพิ่ม.....123
4.111 หน้าจอแสองส่วนประกอบของโปรแกรมแบบฝึกหัด.....124
4.112 หน้าจอ Introduction Exercise.....124
4.113 หน้าจอ แสดงโจทย์ Practice Exercise.....125
4.114 หน้าจอ แสดงกระดาษคำตอบของ Practice Exercise.....125
4.115 หน้าจอ แสดงผลการทดสอบ.....126
4.116 หน้าจอ แสดงประวัติการใช้งานโปรแกรม.....126
4.117 หน้าจอ LogIn สำหรับ Administrator.....127
4.118 หน้าจอแสดงสิทธิ์ที่ Administrator สามารถปฏิบัติการกับระบบได้.....127

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องด้วยการจัดเรียงลำดับข้อมูลให้เป็นระเบียบแบบแผนทำให้สามารถค้นหาและปฏิบัติการกับข้อมูลได้ อย่างสะดวกและรวดเร็ว ดังนั้นการจัดเรียงลำดับข้อมูลจึงมีความสำคัญเป็นอย่างมาก

การจัดเรียงลำดับข้อมูลมีอยู่หลายวิธีซึ่งแต่ละวิธีมีรายละเอียดของขั้นตอนการจัดเรียงที่แตกต่างกันและมีข้อดีข้อเสียที่แตกต่างกันด้วยดังนั้นคณะผู้จัดทำจึงมีความประสงค์ที่จะจัดทำโปรแกรมจำลองการจัดเรียงข้อมูลโดยผ่านเครือข่ายอินเตอร์เน็ตเพื่อให้นักเรียนนักศึกษาหรือผู้ที่สนใจได้ศึกษาและเข้าใจถึงรายละเอียดวิธีการจัดเรียงข้อมูลแต่ละวิธีอีกทั้งได้เล็งเห็นถึงข้อแตกต่างในการจัดเรียงข้อมูลแต่ละวิธีเพื่อความสามารถในการเลือกใช้วิธีการจัดเรียงลำดับให้เหมาะสมกับลักษณะของข้อมูลและจำนวนข้อมูลที่ต้องการจัดเรียง

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อสร้างโปรแกรมจำลองการจัดเรียงลำดับข้อมูลผ่านเครือข่ายอินเตอร์เน็ต

1.2.2 เพื่อให้นักเรียนนักศึกษาหรือผู้สนใจได้มีความเข้าใจถึงรายละเอียดวิธีการจัดเรียงข้อมูลแต่ละวิธีอีกทั้งได้เล็งเห็นถึงข้อแตกต่างในการจัดเรียงข้อมูลแต่ละวิธี

1.2.3 เพื่อให้นักเรียนนักศึกษาหรือผู้สนใจสามารถเลือกใช้วิธีการจัดเรียงลำดับให้เหมาะสมกับลักษณะของข้อมูลและจำนวนข้อมูลที่ต้องการจัดเรียง

1.2.4 เพื่อให้นักเรียนนักศึกษาหรือผู้สนใจสามารถทดสอบความเข้าใจในเนื้อหาการจัดเรียงข้อมูลแต่ละวิธีโดยอาศัยแบบทดสอบ

1.3 สมมุติฐานของการศึกษา

โปรแกรมจำลองการจัดเรียงลำดับข้อมูลผ่านเครือข่ายอินเตอร์เน็ตที่ถูกสร้างขึ้นอย่างสมบูรณ์แล้วจะส่งผลให้ผู้ใช้โปรแกรมสามารถเข้าใจถึงเนื้อหารายละเอียดการเรียงลำดับข้อมูลแต่ละวิธีอีกทั้งสามารถเข้าใจถึงข้อแตกต่างในการจัดเรียงข้อมูลแต่ละวิธีโดยอาศัยการจำลองขั้นตอนการทำงานในแต่ละขั้นตอนด้วยภาพเคลื่อนไหวที่สวยงามน่าสนใจและผู้ใช้โปรแกรมสามารถที่จะเลือกนำวิธีการจัดเรียงข้อมูลที่เหมาะสมกับลักษณะของข้อมูลและจำนวนของ ข้อมูลที่ต้องการจะจัดเรียงไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการศึกษา

รูปแบบของโปรแกรมประกอบด้วยส่วนของเนื้อหาซึ่งได้แก่รายละเอียดขั้นตอนในการเรียงลำดับข้อมูลแต่ละวิธีอีกทั้งข้อดีและข้อเสียของแต่ละวิธีนั้นๆพร้อมทั้งตัวอย่างวิธีการเรียงลำดับของข้อมูลโดยอาศัยภาพเคลื่อนไหวจำลองการเรียงลำดับข้อมูลอย่างเป็นขั้นตอนตามลำดับเพื่อความเข้าใจมากขึ้น พร้อมด้วย ตัวอย่าง Source Code ของโปรแกรมการเรียงลำดับข้อมูลแต่ละวิธีนั้นๆซึ่งจะอ้างอิงกับภาษา Pascalเป็นหลักและส่วนของแบบฝึกหัดเพื่อทบทวนและทดสอบความรู้และความเข้าใจของผู้ใช้โปรแกรม

1.5 ขอบเขตการศึกษา

ปัญหาพิเศษฉบับนี้เป็นโปรแกรมจำลองการจัดเรียงลำดับข้อมูลผ่านเครือข่ายอินเตอร์เน็ตซึ่งจะครอบคลุม เนื้อหาของวิธีการจัดเรียงข้อมูลทั้งหมด 10 วิธีโดยอาศัยหลักการสำคัญทั้งหมด 6 หลักการ ได้แก่ (1) Transposition Sorting method : Bubble Sort , (2) Insert and Keep Sorted method : Insertion Sort , Tree Sort , (3) Priority Queue Sorting method : Selection Sort , Heap Sort , (4) Divide and Conquer method : Quick Sort , Merge Sort , (5) Diminishing Increment Sorting : Shell Sort , (6) Address Calculation Sorting : Proxmap Sort , Radix Sort

1.6 ขั้นตอนการศึกษา

1.6.1 ศึกษาวิธีการเขียนโปรแกรมบน Web

1.6.2 ศึกษาวิธีการใช้ Macromedia FLASH เพื่อใช้ในส่วนของการจำลองขั้นตอนการจัดเรียงลำดับข้อมูลวิธีการต่างๆในลักษณะของภาพเคลื่อนไหว

1.6.3 ศึกษาวิธีการเขียนโปรแกรม ASP เพื่อใช้ในส่วนของแบบฝึกหัด

1.6.4 ศึกษาเนื้อหาวิธีการจัดเรียงข้อมูลวิธีต่างๆ พร้อมทั้งรวบรวม Source Code

1.6.5 จัดทำในส่วนรายละเอียดเนื้อหา , ตัวอย่างและแบบจำลอง และ แบบฝึกหัด

1.6.6 ออกแบบการนำเสนอ

1.6.7 เขียนโปรแกรมและบรรจุเนื้อหา

1.6.8 ทำการทดสอบและแก้ไขข้อบกพร่องของทุกส่วนของโปรแกรม

1.6.9 ปรับแต่งรูปแบบการนำเสนอให้สวยงามและเพิ่มรายละเอียดต่างๆ

1.6.10 จัดทำคู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎีและวรรณกรรมที่เกี่ยวข้อง

2.1 การเขียนโปรแกรมบนเว็บ(Web Programming)

2.1.1 อินเตอร์เน็ตคืออะไร

อินเตอร์เน็ต(Internet)เป็นเครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่มากเกิดจากการเชื่อมต่อเครือข่ายคอมพิวเตอร์จำนวนมากมายในโลกเข้าด้วยกันไม่ว่าจะเป็นเครือข่ายขนาดเล็กเช่นระบบเครือข่ายแบบใกล้หรือระบบเครือข่ายขนาดใหญ่เช่นระบบเครือข่ายมินิหรือเมนเฟรมคอมพิวเตอร์ซึ่งแต่ละเครือข่ายก็จะมีเครื่องคอมพิวเตอร์ที่เป็นเครื่องแม่ข่ายหรือโฮสต์ (Host) ซึ่งมีอยู่หลายชนิดหลายยี่ห้อโดยจะมีการกำหนดข้อตกลงในการสื่อสารเรียกว่า โปรโตคอล (Protocal) ขึ้นมาเพื่อให้คอมพิวเตอร์แต่ละชนิดสามารถติดต่อสื่อสารกันได้โปรโตคอลมาตรฐานที่ใช้ใน การสื่อสารบนอินเตอร์เน็ตจะมีชื่อ เรียกว่า TCP/IP (Transmission Control Protocol/Internet Protocol)

ถ้าเราเปรียบให้โปรโตคอลเหมือนภาษาที่ใช้ในการสื่อสารของมนุษย์ ซึ่งภาษามนุษย์ นั้นมีมากมายหลายภาษาเช่นเดียวกับโปรโตคอล ซึ่งก็จะมีโปรโตคอลอยู่หลายแบบเช่นกัน TCP/IP นั้น จะเปรียบได้กับภาษาอังกฤษ เนื่องจากเป็นโปโตคอลที่เครื่องคอมพิวเตอร์ทุกระบบเข้าใจและมีการใช้ โปรโตคอลนี้ร่วมกันเพื่อการเชื่อมต่อเข้าสู่ อินเตอร์เน็ต

2.1.2 World Wide Web (WWW) คืออะไร ?

World Wide Web หรือเครือข่ายใยแมงมุมคือบริการที่ทำให้ความนิยมในการใช้อินเตอร์เน็ตมีสูงขึ้นเนื่องจากอินเตอร์เน็ตเป็นสิ่งที่ง่ายขึ้นผู้ใช้ไม่จำเป็นต้องรู้คำสั่งของยูนิกซ์อีกต่อไปการอ่านหรือค้นหาข่าวสารข้อมูลทำได้โดยการกดปุ่มเมาส์เพียงอย่างเดียวเอกสารมีการเก็บอยู่ในในลักษณะของสื่อผสมหรือ(Multimedia)ที่เรียกว่า "เว็บเพจ" (Web Page)อันประกอบไปด้วยตัวอักษรข้อความที่มีรูปแบบต่างๆรูปภาพเสียง ภาพเคลื่อนไหววีดีโอและ ไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นการแสดงข้อความที่มีการเชื่อมโยงถึงกัน ซึ่งเป็นลักษณะการเชื่อมต่อที่มีลักษณะ คล้ายกับใยแมงมุมจึงมี การตั้งชื่อบริการนี้ว่าเป็นเครือข่ายใยแมงมุม

2.1.3 ส่วนประกอบของ WWW

1) แหล่งข้อมูลหรือเว็บไซต์ (Web site)เว็บไซต์ หรือเว็บเซอร์ฟเวอร์(Web Server) คือ ระบบคอมพิวเตอร์ที่เป็นแหล่งเก็บเว็บเพจที่ผู้ใช้บริการสามารถเรียกดูเว็บเพจที่อยู่ในเว็บไซต์นั้นได้ซึ่งเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมพิวเตอร์ที่ทำหน้าที่เป็นเว็บไซต์อาจจะใช้ระบบปฏิบัติการยูนิกซ์หรือวินโดวส์เอนทีก็ได้และจะต้องมีการติดตั้งโปรแกรมจัดการที่จะทำให้เครื่องคอมพิวเตอร์เครื่องนั้นทำหน้าที่เป็นเว็บไซต์ ตัวอย่างเช่น ในกรณีที่เครื่องคอมพิวเตอร์นั้นใช้ระบบปฏิบัติการวินโดวส์ เอ็นที จะมีซอฟต์แวร์เว็บเซอร์ฟเวอร์ให้เลือกใช้ เช่น โปรแกรม Purveyor HTTP Server ของสถาบัน Europe Microsoft Windows NT Academic Centre (EMWAC) , โปรแกรม Website การจะใช้บริการ WWW ได้นั้นจำเป็นต้องมีส่วนประกอบสองส่วนคือ

- แหล่งข้อมูลหรือเว็บไซต์(Web site)
- โปรแกรมเว็บบราวเซอร์(Web Browser)

ของบริษัท O'Reilly and Associate หรือถ้าใช้ระบบปฏิบัติการ ยูนิกซ์ก็อาจใช้เว็บเซอร์ฟเวอร์เป็น NCSA httpd , Apache เป็นต้น

ผู้ที่เป็นเจ้าของเว็บไซต์จะจัดสร้างเว็บเพจของตนเก็บไว้ที่เว็บไซต์นั้นเพื่อให้ผู้ใช้คนอื่นๆทั่วโลกสามารถเข้ามาดูเว็บเพจที่เก็บไว้ในเว็บไซต์นั้นได้ ตัวอย่างเช่น เว็บเพจของสำนักข่าว CNN จะเก็บอยู่ที่เว็บไซต์ www.cnn.com เป็นต้น

2) โปรแกรมเว็บบราวเซอร์ (Web Browswer) เป็นโปรแกรม หรือ ซอฟต์แวร์ที่ใช้ในการ เข้าสู่ WWW และเปิดดูเว็บเพจที่เก็บอยู่ในเว็บไซต์โปรแกรมเว็บบราวเซอร์ส่วนมากทำงานร่วมกับโปรแกรมวินโดวส์และนอกจากจะใช้เพื่อดูเว็บเพจจากเว็บไซต์ใดๆแล้วหลายโปรแกรมยังมีความ สามารถอื่นๆ เช่น บริการส่งจดหมายอิเล็กทรอนิกส์ การค้นหาข้อมูล การโอนถ่ายโปรแกรมด้วย FTP เป็นต้น

2.1.4 IP Address , DNS , E-mail Address และ URL คืออะไร ?

1) IP Address

เนื่องจากในระบบเครือข่ายอินเตอร์เน็ตจะใช้โปรโตคอล TCP/IP เป็นมาตรฐานในการสื่อสาร ข้อมูลซึ่งจะมีการกำหนดหมายเลขประจำตัวที่ไม่ซ้ำกันให้กับเครื่องคอมพิวเตอร์ทุกเครื่องที่มีการเชื่อมกันอยู่ในระบบ เครือข่ายหมายเลขประจำตัวนี้จะถูกเรียกว่า IP Address หรือหมายเลข IP โดยมีรูปแบบเป็นชุดของตัวเลข 4 ชุด ที่คั่นด้วยเครื่องหมายจุด เช่น 202.44.192.43 ตัวเลขในแต่ละชุดจะมีขนาด 8 บิต ดังนั้นในแต่ละชุดจะมีค่าได้ตั้งแต่ 0 ถึง 2^8-1= 255 เท่านั้น ซึ่งทำให้สามารถกำหนดหมายเลข IP ให้กับเครื่องคอมพิวเตอร์ได้ทั้งหมดถึง 4 พันล้านเลขหมายที่ไม่ ซ้ำกันเลย

ดังนั้นเมื่อมีการติดตั้งเครื่องคอมพิวเตอร์ที่เป็นโฮสต์คอมพิวเตอร์เพื่อเชื่อมต่อเข้ากับอินเตอร์เน็ตและให้บริการต่างๆจำเป็นต้องขอหมายเลข IP จากหน่วยงาน Internet Network Information Center (InterNIC) ขององค์กร Network Solution Incorporated(NSI) ที่รัฐเวอร์จิเนีย สหรัฐอเมริกา แต่ถ้าผู้ใช้เป็นสมาชิกกับหน่วยงาน ที่ทำหน้าที่เป็นผู้ให้บริการอินเตอร์เน็ต (Internet Service Provider : ISP) ก็ไม่ต้องขอหมายเลข IP เนื่องจาก ISP จะเป็นผู้ส่งหมายเลข IP ให้แก่ผู้ใช้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) DNS(Domain Name System)

การใช้หมายเลข IP ซึ่งเป็นตัวเลขล้วนๆ ในการอ้างอิงถึงเครื่องคอมพิวเตอร์แต่ละเครื่องนั้นจะ มีข้อเสียคือจำยากและก่อให้เกิดความสับสนได้ง่ายจึงมีการพัฒนาวิธีการอ้างอิงถึงหมายเลข IP แบบใหม่ที่เรียกว่า Domain Name System(DNS) ขึ้นมา

DNS เป็นเทคนิคการเปลี่ยนหมายเลข IP ที่เป็นตัวเลขให้เป็นตัวอักษรแทนเช่น หมายเลข IP เป็น 202.12.97.1 ผู้ใช้สามารถเขียนเป็นชื่อโดเมนคือ kku1.kku.ac.th แทน เครื่องคอมพิวเตอร์ที่มีชื่อว่า Domain Name Server จะทำหน้าที่ในการแปลงจากชื่อโดเมนให้เป็นหมาย เลข IP อีกทีหนึ่งรูปแบบของชื่อโดเมนมีดังต่อไปนี้

ชื่อโฮสต์คอมพิวเตอร์.ชื่อเครือข่ายท้องถิ่น[.ชื่อโดเมนย่อย.].ชื่อโดเมนระดับบนสุด

ชื่อโดเมนจะมีการแบ่งออกเป็นระดับชั้นโดยใช้เครื่องหมายจุดคั่น ชื่อโดเมนที่อยู่ทางด้านขวาสุดจะเรียกว่าชื่อโดเมนระดับบนสุด ซึ่งจะแบ่งออกได้เป็น 2 ประเภทคือ

1. โดเมนระดับบนสุดที่เป็นชื่อย่อของประเภทขององค์กรในสหรัฐอเมริกา เช่น

Com	commercial	กลุ่มองค์กรเอกชน
edu	educational	กลุ่มสถาบันการศึกษา
gov	government	กลุ่มองค์กรของรัฐทั่วไป
mil	military	กลุ่มองค์กรทหาร
net	networks services	กลุ่มองค์กรบริการเครือข่าย
org	non-commerial organization	กลุ่มองค์กรที่ไม่แสวงหาผลกำไร

ตัวอย่างเช่น ชื่อโดเมน biz.zd.com โดเมนระดับบนสุดคือ com ที่แสดงถึงองค์กรเอกชน ที่อยู่ในประเทศสหรัฐอเมริกา สำหรับระดับถัดไปคือ zd จะแสดงถึงชื่อเครือ ข่ายของบริษัทเอกชนนั้น และ biz จะหมายถึงชื่อโฮสต์ที่ให้บริการเป็นต้น

2. ชื่อโดเมนระดับบนสุดที่เป็นชื่อย่อของประเทศต่างๆเช่น

au	ออสเตรเลีย	, jp	ญี่ปุ่น
ca	แคนาดา	, th	ไทย
fr	ฝรั่งเศษ	, uk	อังกฤษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีสับโดเมน(Subdomain) ที่แสดงถึงประเภทขององค์กรในประเทศนั้นๆเช่น

ac	สถาบันการศึกษา	go	องค์กรรัฐบาล
co	องค์กรเอกชน	or	องค์กรไม่แสวงหาผลกำไร

ตัวอย่างเช่น ชื่อโดเมน kku1.kku.ac.th มีโดเมนระดับบนสุดคือ th ที่หมายถึงประเทศไทย ระดับรองลงมาคือโดเมนย่อย ac หมายถึงเป็นสถาบันการศึกษา ระดับถัดไปคือ kku หมายถึง มหาวิทยาลัยขอนแก่นซึ่งโดยทั่วไปจะหมายถึง เครือข่ายมหาวิทยาลัยขอนแก่น

3) Email Address

ชื่อโดเมนได้ถูกนำมาใช้สร้างที่อยู่ของจดหมายอิเล็กทรอนิกส์หรืออีเมลล์เพื่อผู้ใช้งานอินเตอร์เน็ตสามารถรับส่งจดหมายกันได้ ซึ่งผู้ใช้งานแต่ละคนต้องมีที่อยู่อีเมลล์ดังต่อไปนี้

ชื่อผู้ใช้@ชื่อโฮสต์.ชื่อเครือข่ายท้องถิ่น[.ชื่อโดเมนย่อย.].ชื่อโดเมนระดับบนสุด

4) URL (Uniform Resource Locator)

ซึ่งโปรแกรมเว็บบราวเซอร์จะใช้ข้อมูล URL นี้ในการค้นหาที่อยู่ของเว็บไซต์ที่ผู้ใช้ บริการจะไปเรียกดูข้อมูล รูปแบบของ URL มีดังนี้

โปรโตคอล :// ชื่อโดเมน/ไดเรกทอรีที่เก็บไฟล์ในโฮสต์/ชื่อไฟล์ในโฮสต์

2.2 แบบจำลองคืออะไร

แบบจำลองหมายถึง ตัวแทนของวัตถุ ระบบ หรือแนวคิดลักษณะใดลักษณะหนึ่งแบบจำลองอาจนำไปใช้ งาน ในหลายลักษณะดังนี้

1. เป็นเครื่องช่วยคิด (An aid to thought) เช่นแบบจำลองโครงข่าย (Network Model) ช่วยทำให้ผู้สร้างแบบจำลองได้มองเห็นว่าจะมีกิจกรรมที่ต้องทำอะไรบ้างและทำอะไรก่อนอะไรหลัง

2. เป็นเครื่องสื่อความหมาย (An aid to communication) แบบจำลองจะช่วยให้เข้าใจพฤติกรรมของระบบงานและช่วยให้สามารถอธิบายพฤติกรรม ปัญหา และการแก้ปัญหาของระบบงาน

3. เป็นเครื่องช่วยสอนและฝึกอบรม (Purposes of training and instruction) เช่น แบบจำลองควบคุมการบินจะช่วยให้นักบินทำความเข้าใจและคุ้นเคยกับระบบการควบคุมเครื่องบินจริงก่อนขึ้นฝึกบินจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เป็นเครื่องมือสำหรับการทำนาย (A tool of prediction) จากการที่แบบจำลองจะช่วยให้เข้าใจพฤติกรรมของระบบงาน ก็จะช่วยให้ผู้สร้างแบบจำลองสามารถคาดคะเนหรือทำนายได้ว่าเมื่อมีเหตุการณ์ที่มีผล กระทบต่อองค์ประกอบของระบบเกิดขึ้น จะมีผลอะไรเกิดขึ้นกับระบบ

5. เป็นเครื่องมือสำหรับการทดลอง (An aid to experimentation) โดยที่แบบจำลองเป็นสิ่งที่สร้างขึ้นแทนระบบจริงในกรณีที่ต้องการศึกษาเงื่อนไขต่างๆกับระบบงานจริงแต่ทำไม่ได้ก็น่าจะเอาเงื่อนไขนั้นๆมาทดลองกับแบบจำลองเพื่อดูว่าจะให้ผลอย่างไรเพื่อประโยชน์ในการตัดสินใจว่าควรจะนำเงื่อนไขนั้นๆไปใช้กับระบบ งานจริงหรือไม่

2.3 การหาฟังก์ชัน Big-O

ฟังก์ชัน Big – O สามารถหาได้จากจำนวนครั้งการเปรียบเทียบใน Algorithm นั้นๆ จาก $g(n) = O(f(n))$ โดยจะมีค่าคงที่ k และ n ซึ่ง $g(n) <= k|f(n)|$ สำหรับทุกๆ $n >= n_0$

ตัวอย่างที่ 1

หาฟังก์ชัน O ของ $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ จะได้ว่า $\sum_{i=1}^{n} i = \frac{(n-1)(n-1+1)}{2} = \frac{n^2}{2} - \frac{n}{2}$

ตารางที่ 2.1

n	$\frac{n^2-n}{2}$	$\frac{n^2}{2}$	N^2
1	0	0.5	1
2	1	2	4
3	2	4.5	9

จะเห็นว่า $\frac{n(n+1)}{2} <= \frac{n}{2}$ ทุกๆ $n >= n_0$

เมื่อ $k = \frac{1}{2}$, $n_0 = 1$

ดังนั้น $\frac{n(n+1)}{2} = O(n^2)$

ตัวอย่างที่ 2

หาฟังก์ชัน O ของ $\frac{n(n+1)}{2}$ จะได้ว่า $\frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2

n	$\frac{n(n+1)}{2}$	$\frac{n}{2}$	N^2
1	1	0.5	1
2	3	2	4
3	6	4.5	9

จะเห็นว่า $\frac{n(n+1)}{2} <= n$ ทุกๆ $n >= n_0$

เมื่อ $k = 1$ และ $n_0 = 1$

ดังนั้น $\frac{n(n+1)}{2} = O(n^2)$

<u>ตัวอย่างที่ 3</u>

หาฟังก์ชัน O ของ $\sum_{i=1}^{n} 3i$ จะได้ว่า $\sum_{i=1}^{n} 3i = \frac{3n(n+1)}{2} = \frac{3n^2}{2} + \frac{3n}{2}$

ตารางที่ 2.3

N	$\frac{3n(n+1)}{2}$	3n	$\frac{3n^2}{2}$	n
1	3	3	1.5	1
2	9	12	6	4
3	18	27	13.5	9

จะได้ว่า $\frac{3n(n+1)}{2} <= 3n^2$ ทุกๆ $n >= n_0$

เมื่อ $k=3$, $n_0 = 1$

ดังนั้น $\frac{3n(n+1)}{2} = O(n^2)$

<u>ตัวอย่างที่ 4</u>

หาฟังก์ชัน O ของ $\frac{n-1}{2}$ จะได้ว่า $\frac{n-1}{2} = \frac{n}{2} - \frac{1}{2}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4

n	$\frac{n-1}{2}$	$\frac{n}{2}$
1	0	0.5
2	0.5	1
3	1	1.5

จะได้ว่า $\frac{(n-1)}{2} <= \frac{n}{2}$ สำหรับทุกๆ $n >= n_0$

เมื่อ $k= \frac{1}{2}$, $n = 1$

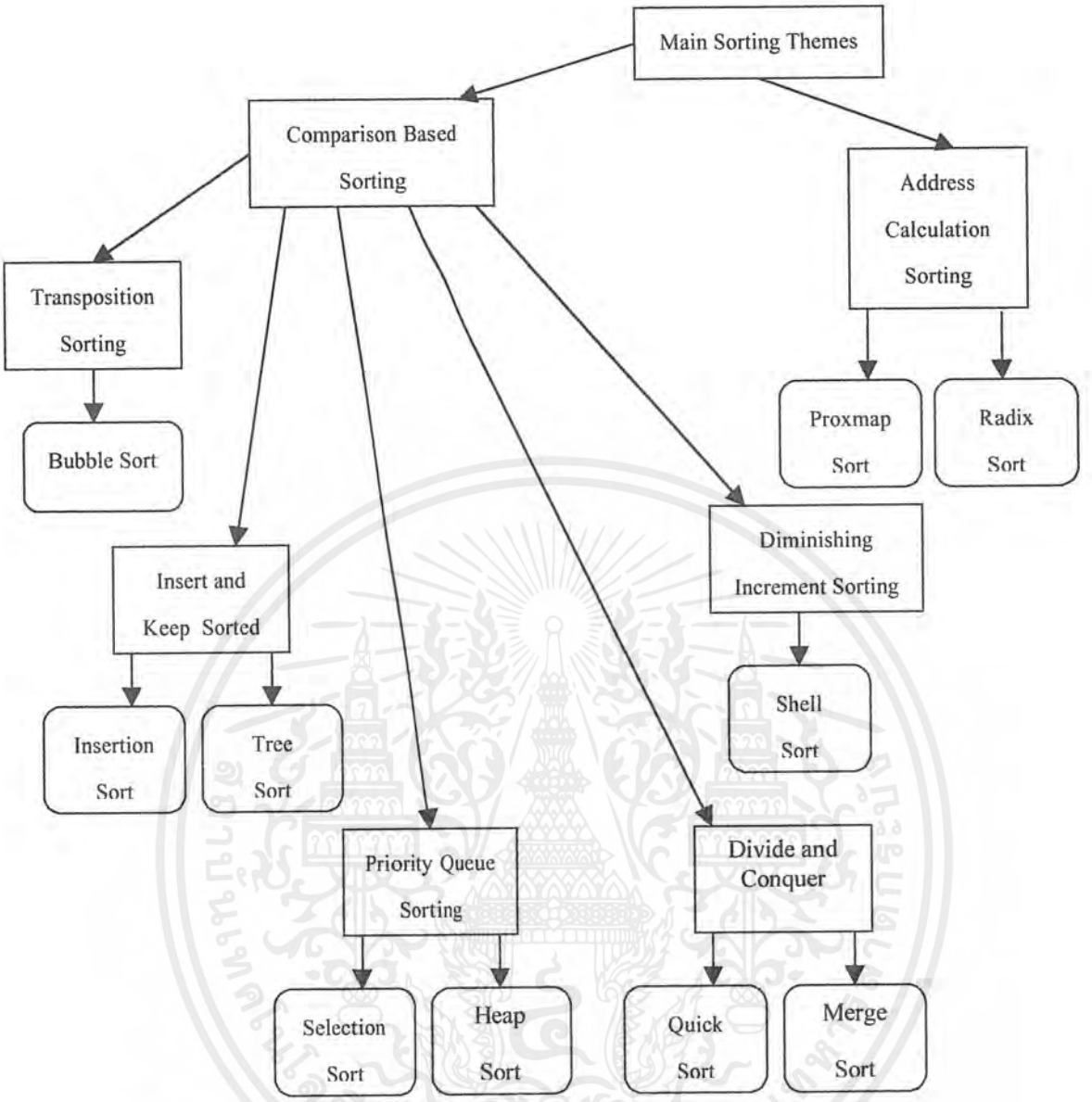
จะได้ $\frac{n-1}{2} = O(n)$

2.4 การเรียงลำดับ (Sorting)

ปัญหาของการเรียงลำดับเริ่มต้นที่กลุ่มของข้อมูลที่ไม่ได้มีการจัดเรียงกลุ่มหนึ่ง ซึ่งการจัดเรียงลำดับข้อมูลกลุ่มนี้สามารถกระทำได้หลายวิธีโดยที่แต่ละวิธีมีประสิทธิภาพที่แตกต่างกัน วิธีการจัดเรียงบางประเภทสามารถทำงานได้ดีภายในหน่วยความจำหลักแต่ไม่สามารถทำงานได้ดีกับงานที่มีขนาดใหญ่ที่ถูกเก็บไว้ในหน่วยความจำสำรอง เช่น เทป

วิธีการจัดเรียงลำดับสามารถแบ่งได้หลายประเภทโดยอาศัยการจับกลุ่มของพื้นฐานการทำงานที่คล้ายคลึงกันได้ดังภาพที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 Sorting Themes

2.4.1 Priority Queue Sorting Methods

2.4.1.1 Selection Sort

การเรียงลำดับด้วยวิธีนี้มีหลักการดังนี้

1. เลือกค่าที่น้อยที่สุด
2. และนำมาแลกเปลี่ยนลำดับกับค่าใน a[1]

(ถ้าชุดของข้อมูลเป็นลิสต์ ประกอบด้วย a[1] , a[2] , . . . ,a[n])

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นก็กระทำตามหลักการทั้ง 2 กับข้อมูลที่เหลือ คือในครั้งที่ 2 ค่า a[2] ถูกแลกเปลี่ยนกับค่าที่เลือกแล้วว่าน้อยที่สุดในลิสต์ a[2] ... a[n] และในครั้งที่ 3 ค่า a[3] ถูกแลกเปลี่ยนกับค่าที่น้อยที่สุดใน a[3] ... a[n] และเรื่อยไปจนเหลือสมาชิกในลิสต์สองค่าคือ a[n-1] และ a[n] ดังนั้นจำนวนรอบในการกระทำเป็น n-1 รอบ

ขั้นตอนในการเรียงลำดับแบบการเลือกนี้ เขียนเป็นโปรซีเจอร์ Selectionsort (a,n) ดังภาพที่ 2.2

```
PROCEDURE  selectionsort (VAR a : arrtype ; n : index);
VAR   i,j,k    :      index;
      Min      :      itemtype;
      BEGIN
        FOR i:=1   TO    n-1    DO
        BEGIN
            k     :=     i      ;
          min     :=     a[i]   ;   {k = position of the smallest value of the rest list}
          FOR  j:=i+1  TO    n      DO
          IF   a[j]    <     min    THEN
          BEGIN
              k     :=     j;
              min   :=     a[j];
          END;
          a[k]  :=     a[i];
          a[i]  :=     min; {exchange  value between min and a[i] }
        END; END;
```

ภาพที่ 2.2 โปรซีเจอร์ Selectionsort

ภาพที่ 2.3 แสดงขั้นตอนการใช้อัลกอริทึมในโปรซีเจอร์ Selectionsort (a,n) กับข้อมูล 5 20 7 10 13 2 30 ตัวอักษรสีน้ำเงินแทนตัวเลขที่มีค่าน้อยที่สุดในลิสต์ที่เหลือ ส่วนเส้นตรง (_____) ที่ขีดไว้แสดงถึงชุดของข้อมูล ที่เหลือสังเกตว่าสมาชิกแรกคือ a[i] ในลิสต์ของข้อมูล a[i]...a[n] ที่เป็นสมาชิกที่มีค่าน้อยที่สุดเสียเองมี 7 10 13 และ 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเดิม	5	20	7	10	13	2	30
i=1	5	20	7	10	13	2	30
i=2	2	20	7	10	13	5	30
i=3	2	5	7	10	13	20	30
i=4	2	5	7	10	13	20	30
i=5	2	5	7	10	13	20	30
i=6	2	5	7	10	13	20	30

ภาพที่ 2.3 ขั้นตอนการใช้อัลกอริทึม Selectionsort (a,n) กับข้อมูล 5 20 7 10 13 2 30

ประสิทธิภาพของ Selection Sort เมื่อวิเคราะห์จำนวนครั้งการเปรียบเทียบของการเรียงลำดับด้วยการเลือก ตามอัลกอริทึมในโปรซีเจอร์ Selectionsort (a,n) เห็นว่า $C_{min} = C_{max}$ ซึ่งเท่ากับ

$$(n-1)+(n-2)+\ldots 1 = \frac{n(n-1)}{2} = \frac{(n^2-n)}{2}$$

ซึ่งเขียนได้ว่า $C = \frac{(n^2-n)}{2} = O(n^2)$

2.4.1.2 Heap Sort

การสร้างต้นไม้ฮีปมี 2 ขั้นตอนคือ

1. รับข้อมูลเข้าสร้างเป็นต้นไม้
2. ปรับต้นไม้ให้มีคุณสมบัติเป็นต้นไม้ฮีปซึ่งหมายถึง $h_i >= h_{2i}$ และ $h_i >= h_{2i+1}$ (I= 1, 2 , . . . ,n / 2)

หลังจากได้ต้นไม้ฮีปแล้ว ก็สามารถทำการเรียงลำดับด้วยวิธี ฮีปซอร์ทได้ ซึ่งมีอัลกอริทึมดังนี้

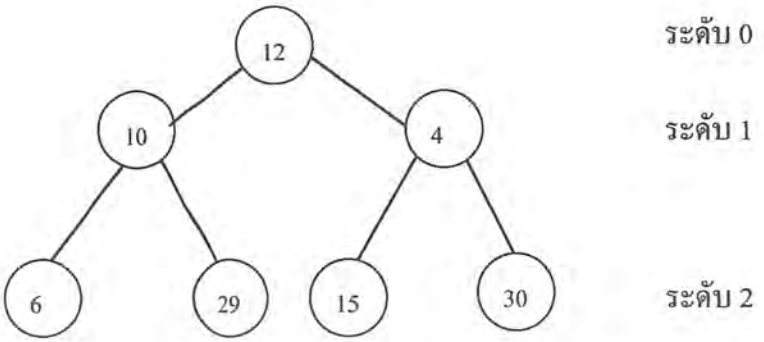
1. r <- - n (* r เป็นตัวแปรแสดงจำนวนข้อมูลในต้นไม้โดยครั้งแรกเซตเป็น n *)
2. ในระหว่างที่ r > 1 ทำ
   2.1 สับเปลี่ยนค่าใน h_1 และ h_r
   2.2 r <- - r — 1 (* ลดจำนวนข้อมูลลง 1 *)
   2.3 ปรับต้นไม้ที่เหลือ $h_1 , h_2 \ldots , h_r$ ให้เป็นฮีป (* โดยปรับที่เส้นจากเริ่มจาก h_1 เท่านั้น เนื่องจาก $h_2 , h_3 , \ldots h_r$ มีคุณสมบัติเป็นฮีปอยู่แล้ว *)
   2.4 จบงาน

สรุปได้ว่าการทำฮีปซอร์ท มีขั้นตอนใหญ่ๆ อยู่ 2 ขั้นตอน คือ

1. สร้างต้นไม้ฮีปโดยการปรับค่าในโหนดที่ไม่ใช่ลีฟ ตามเส้นทางไปจนถึงลีฟโหนด
2. นำค่าใน h_1 เป็นผลลัพธ์ และปรับค่าใน h_1 ในเส้นทางไปจนถึงลีฟโหนด

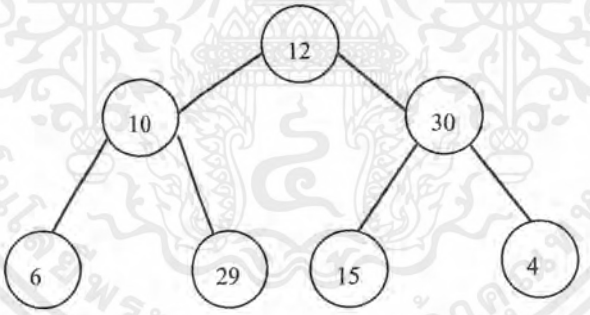
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังเช่นมีชุดของข้อมูลเข้าเป็น 12 10 4 6 29 15 30 เมื่อนำมาสร้างเป็นต้นไม้ได้ดังภาพที่ 2.4 ซึ่งมีระดับสูงสุดเป็น 3

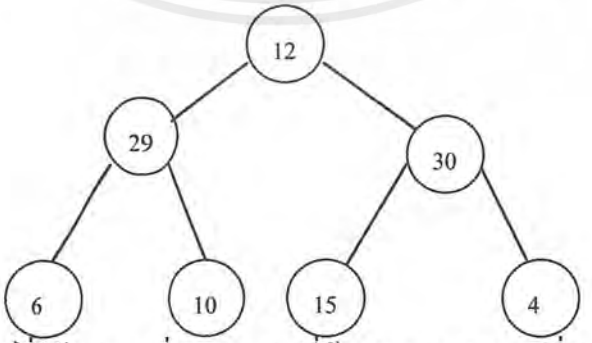


ภาพที่ 2.4 ต้นไม้ที่ได้จากชุดข้อมูล 12 10 4 6 29 15 30

ซึ่งภาพต้นไม้ ภาพที่ 2.4 ยังไม่มีคุณสมบัติเป็นต้นไม้ฮีปที่ว่า $h_i >= h_{2i}$ และ $h_i >= h_{2I+1}$ ($i = 1,2,\ldots,n/2$) คือค่าที่อยู่ในโหนดพ่อมีค่ามากกว่าในโหนดลูกทั้งสอง (ถ้าโหนดนั้นมีลูก) จึงต้องปรับต้นไม้ให้มีคุณสมบัติเป็นฮีป โดยการเปลี่ยนค่า h_i ที่ไม่ใช่ลีฟโหนด ($2i < n$) ด้วยค่าของ h_{2i} หรือ h_{2i+1} ไปตามเส้นทางของลูกซ้ายหรือขวา ขึ้นอยู่กับว่านำค่าของลูกซ้ายหรือลูกขวาไปแทนที่ใน h_j ทำเช่นนี้ไปจนถึงลีฟโหนด (ค่า i จะเปลี่ยนเป็น 2i หรือ 2i+1) ซึ่งจากภาพที่ 2.4 โหนดที่ไม่ใช่ลีฟคือ $h_1 h_2 h_3$

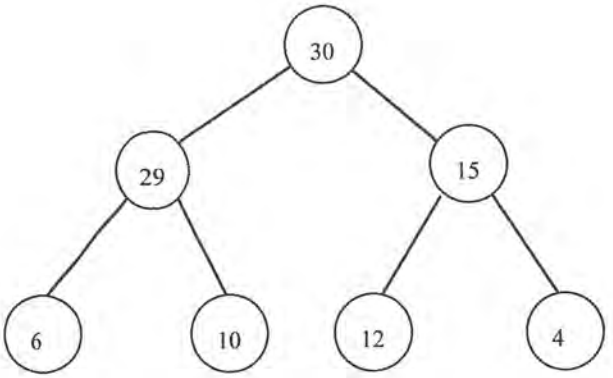


ภาพที่ 2.5 ผลลัพธ์ที่ได้จากการที่ h_3 ในภาพที่ 2.4 ถูกแทนที่ด้วย h_7 และ h_7 ถูกเปลี่ยนค่าเป็น h_3



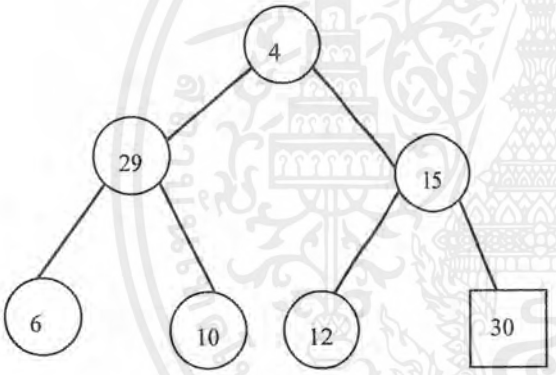
ภาพที่ 2.6 ผลลัพธ์ที่ได้จากการที่ h_2 ถูกแทนที่ด้วย h_5 และ h_5 ถูกเปลี่ยนค่าเป็น h_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

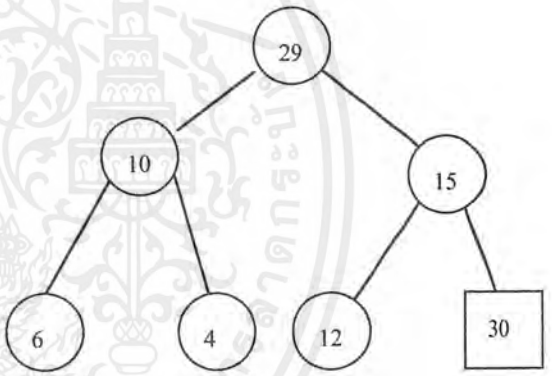


ภาพที่ 2.7 h_1ถูกแทนที่ด้วยh_3 ซึ่งมากกว่า h_2 และมากกว่า h_1ดังนั้น h_3 มีค่าใหม่เป็น 12 แต่ถูกแทนด้วย h_6 ซึ่งมากกว่า h_7 และ h_3

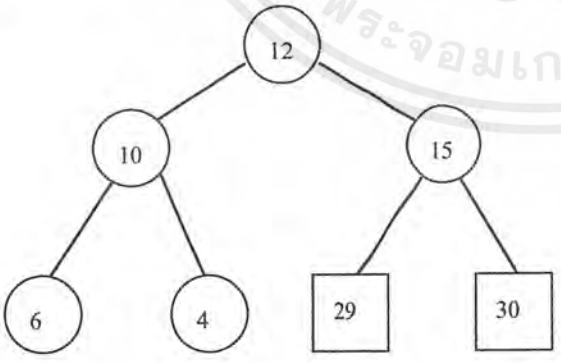
และจากต้นไม้ฮีปในภาพที่ 2.7 นำมาเรียงลำดับตามอัลกอริทึมฮีปซอร์ท ได้ดังภาพที่ 2.8 ถึง ภาพที่ 2.13 โดยภาพ a) แสดงการสับเปลี่ยน h_1 และ h_r ส่วนภาพ b) แสดงต้นไม้ฮีปหลังปรับเปลี่ยนแล้ว



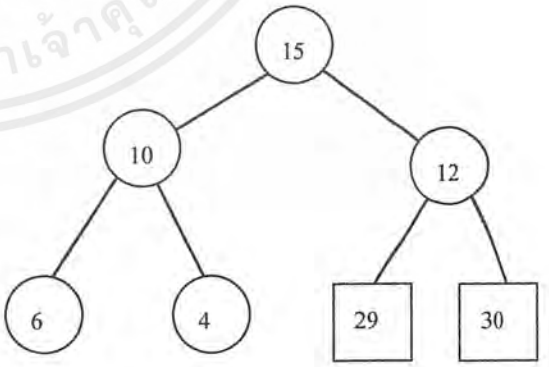
ภาพที่ 2.8 a



ภาพที่ 2.8 b

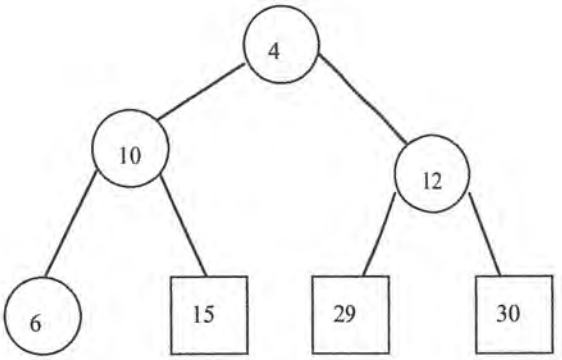


ภาพที่ 2.9 a

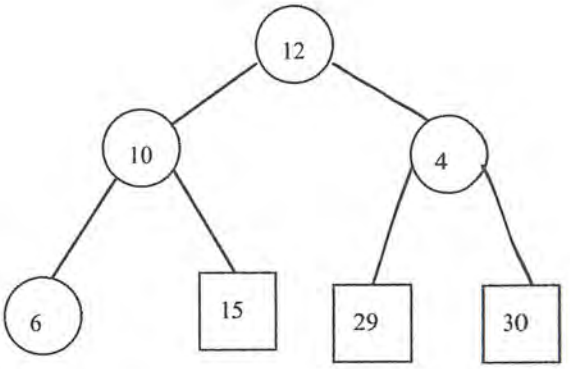


ภาพที่ 2.9 b

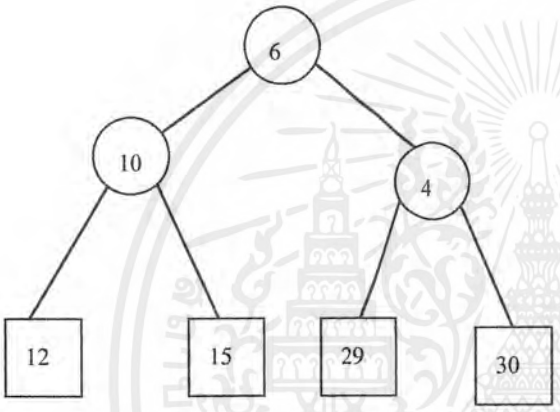
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



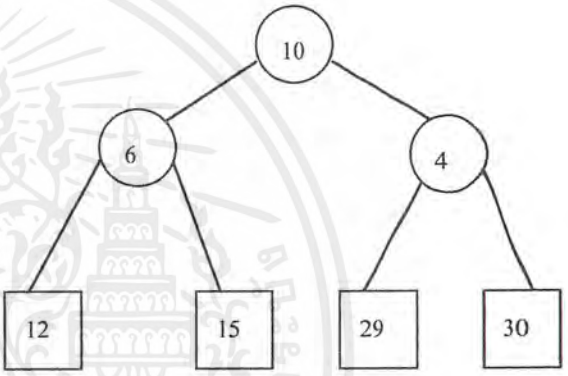
ภาพที่ 2.10 a



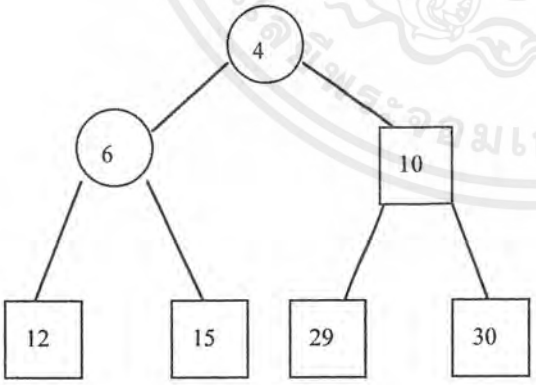
ภาพที่ 2.10 b



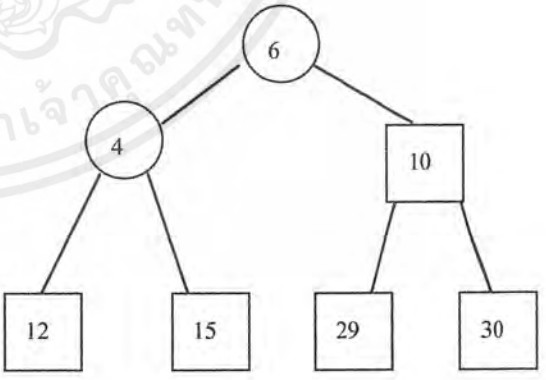
ภาพที่ 2.11 a



ภาพที่ 2.11 b

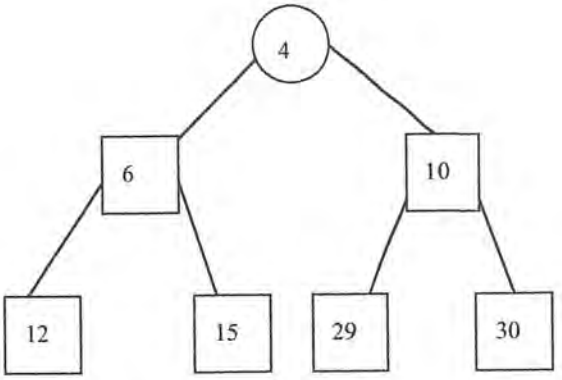


ภาพที่ 2.12 a

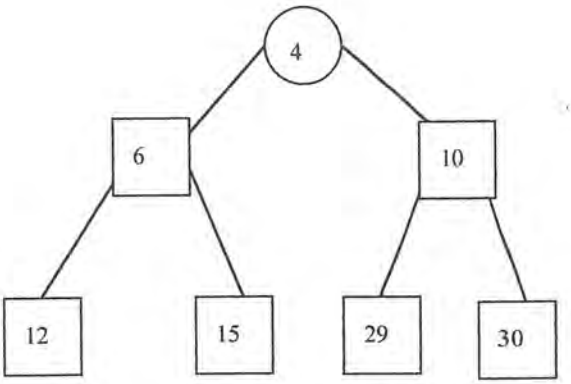


ภาพที่ 2.12 b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



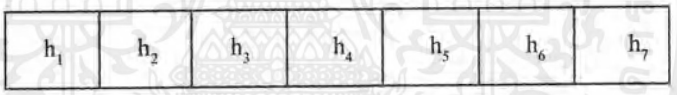
ภาพที่ 2.13 a



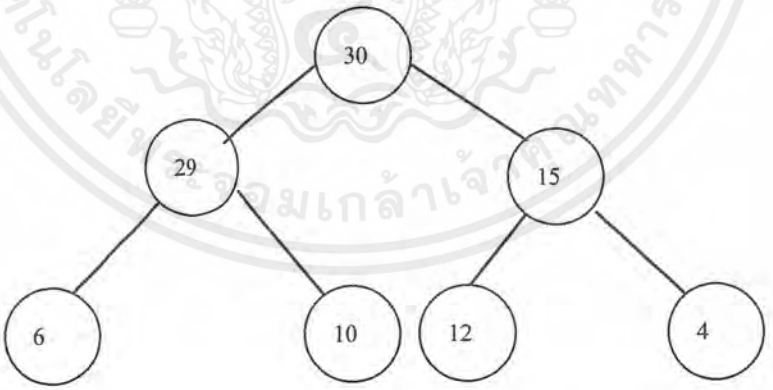
ภาพที่ 2.13 b

การแทนที่ต้นไม้ฮีปด้วยอาร์เรย์

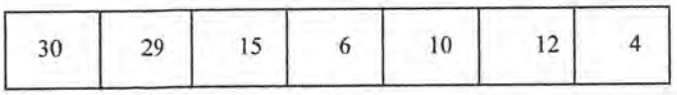
จากขั้นตอนการทำฮีปซอร์ทพบว่าการเข้าถึงข้อมูลทำได้ 2 ลักษณะคือ จากพ่อแล้วไปหาลูก เช่น จาก h_i และ ไปยัง h_{2i} และ h_{2i+1} หรืออีกลักษณะคือไปยังลำดับข้อมูลที่มาก่อน เช่น จาก h_i ไปยัง h_{i-1} ดังนั้นการแทนต้นไม้ฮีป ด้วยอาร์เรย์จึงเป็นทางเลือกที่ดีและเหมาะสม ซึ่งลำดับของข้อมูลในต้นไม้ฮีปจะเป็นลำดับที่ในอาร์เรย์ทำให้เขียน โปรซีเจอร์ทำฮีปซอร์ทได้ง่ายกว่าเมื่อแทนด้วยลิงค์ เช่น ถ้าเราจองอาร์เรย์ h เป็น ARRAY[1..n] OF itemtype ไว้ อาร์เรย์ h เป็นดังภาพที่ 2.14



ภาพที่ 2.14



ภาพที่ 2.15



ภาพที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ภาพที่ 2.16 เป็นอาร์เรย์แทนต้นไม้ฮีปของภาพที่ 2.15 ซึ่ง ณ h[i] ใด ๆ สามารถไปยังโหนดพ่อด้วยการหาร i ด้วย 2 แล้วปัดเศษทิ้ง (i DIV 2) และถ้าต้องการไปยังโหนดลูกซ้ายหรือขวาคือไปยัง h [2* i] หรือ h [2 * i +1] นั่นเอง

โปรแกรม Heap Sort ภาพที่ 2.17 รับอาร์เรย์ของต้นไม้ และทำการปรับให้เป็นต้นไม้ฮีปโดยใช้โปรซีเจอร์ Adjheap (h,L,r) หลังจากนั้นทำการเรียงลำดับข้อมูลคือนำค่าในรู้ตโหนดออกเป็นผลลัพธ์แล้วใช้ Adjheap (h,1,r) ปรับต้นไม้เป็นฮีป ณ เส้นทางจาก h[1] ลงไปยังลีฟโหนด และทำเช่นเดิมเรื่อยๆ จนเหลือข้อมูลที่รู้ตเพียงค่าเดียว ก็ถือว่าการเรียงลำดับสิ้นสุดแล้ว ภาพที่ 2.18 แสดงผลการทำงานของโปรแกรมเมื่อคีย์ข้อมูลชุดนี้คือ 8 17 10 60 20 15 50 4 65

```
PROGRAM heapsort ( input , output )
USES        Crt;
CONST           max = 20 ;
TYPE        itemtype = integer ;
            heap     = 1..max ;
VAR         h        :    heap ;
            x        :    itemtype;
             i,j,k,L,r       :       index ;
            n ,order :    index ;
            tap      :    integer ;
PROCEDURE adjheap (VAR h : heap ; L , r : index ) ;
LABEL       replace ;
VAR         i,j      :    index;
            X        :    itemtype ;
BEGIN
      i := L;  j := 2* i ; X := h [ i ] ;
      WHILE j <= r DO
      BEGIN IF j < r THEN
            IF h [j] < h [j+1] THEN j := j+1 ;
               IF X >= h[j] THEN GOTO replace;
            (*ELSE*)
            h [ i ] := h [j] ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

39674

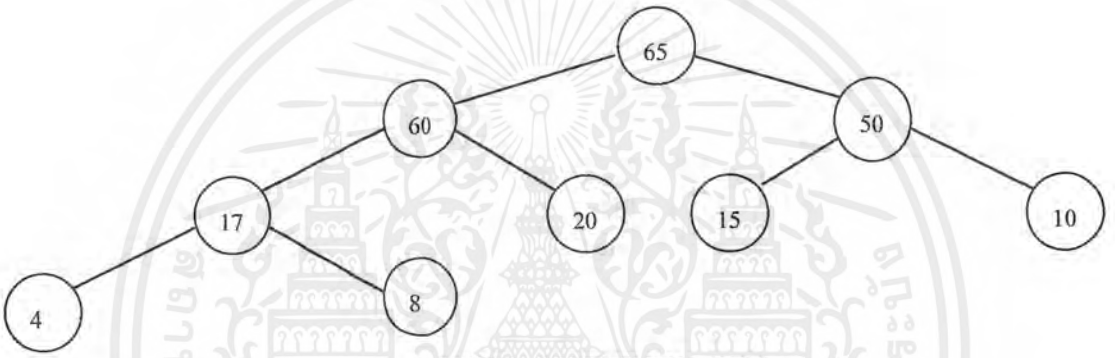
```
                i : = j; j : = 2 * i ;
        END;
replace   h [ i ] = X;
END;
PROCEDURE  printree  ( h : heap ; order : index ; tap : integer ) ;
VAR    i : index;
BEGIN
        tap  : = tap + 1 ;
        IF  order  <= n  THEN
        BEGIN
                printree ( h , order * 2 + 1 , tap );
                FOR  i : = 1  TO  tap  DO  WRITE (' ');
                WRITELN  ( h [order ] );
                printree  ( h , order * 2 ,tap );
        END;
END;
BEGIN  ( * sort * )
        WRITELN ('ENTER DATA , 999 FOR END ');
        n := 0 ;  tap := 2 ; order : = 1;
        READ (X);
        WHILE  X <> 999 DO
        BEGIN
                n := n+1 ; h[n] := X ;
                READ (X);
        END;
L := ( n div 2 )  +  1 ;  r  :=  n;
WHILE  L>1  DO
BEGIN  (* exchange  h[1] ,h[r] AND adjust  tree * )
        X := h[1] ; h[1] := h[r] ; h[r] := X;
        r := r-1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        adjheap (h,1,r);
    END;
    WRITELN;
    WRITELN(' OUTPUT ASCENDING KEY ');
    FOR I := 1 TO n DO WRITE (h[I] :3 );
    END.
```

ภาพที่ 2.17

```
ENTER DATA , 999  FOR STOP
 8    17    10    60    20    15    50    4    65    999
AFTER ADJUST TO HEAP TREE
```




```
OUTPUT ASCENDING KEY
 4    8    10    15    17    20    50    60    65
```

ภาพที่ 2.18

สังเกตโปรซีเจอร์ printree (h,order,tap) ทำงานอย่างรีเคอร์ซีฟ ทำการพิมพ์อาร์เรย์ที่เก็บต้นไม้ฮีป ให้เป็น ต้นไม้โดยต้นไม้ย่อยขวาอยู่บนสุด รู้ตอยู่กลางและต้นไม้ย่อยซ้ายอยู่ด้านล่าง

ในกรณีที่แย่ที่สุดมีการปรับข้อมูล n ตัว เส้นทางที่ปรับเป็น $\log_2 n$ (ความสูงของต้นไม้) ดังนั้นประสิทธิภาพของฮีปซอร์ทในกรณีที่แย่ที่สุดเป็น $O(n\log_2 n)$

2.4.2 Divide — and — Conquer Methods

โดยทั่วไปการแก้ปัญหาต่าง ๆจะใช้เวลามากขึ้นเมื่อขนาดของข้อมูลใหญ่ขึ้น เช่นในกรณีของ Bubble Sort มีTime Complexity คือ $O(n^2)$

- จำนวนข้อมูล 100 ตัว มีจำนวนการเปรียบเทียบ $\sum_{i=1}^{100} i = \frac{100*(99)}{2} = 4{,}950$ ครั้ง

$$\sum_{i=1}^{200}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จำนวนข้อมูล 200 ตัว มีจำนวนการเปรียบเทียบ $\sum_{i=1} i = \frac{200 * (199)}{2} = 19,900$ ครั้ง

ดังนั้นจึงมีแนวคิดในการแก้ปัญหาโดยแบ่งข้อมูลเป็นหลายๆกลุ่ม แล้วจึงทำการเรียงลำดับข้อมูลในกลุ่ม ข้อมูลดังกล่าวให้ถูกต้อง แล้วจึงนำกลุ่มข้อมูลดังกล่าวมารวมกัน (Merge) จนเป็นข้อมูลชุดเดิมก่อนการแบ่งกลุ่ม หลักการแบบนี้เรียกว่า Divide and Conquer

- จำนวนข้อมูล 400 ตัว แบ่งเป็น 2 กลุ่ม กลุ่มละ 200 ตัว

จำนวนครั้งการเปรียบเทียบคือ $\sum_{i=1}^{200} i + \sum_{i=1}^{200} i = 19,900 + 19,900 = 39800$ ครั้ง

จำนวนครั้งของการ Merge คือ 1 ครั้ง

- จำนวนข้อมูล 400 ตัว แบ่งเป็น 4 กลุ่ม กลุ่มละ 100 ตัว

จำนวนครั้งการเปรียบเทียบคือ $4 * \sum_{i=1}^{100} i = 4 * 4,950 = 19,800$ ครั้ง

จำนวนครั้งของการ Merge คือ 3 ครั้ง

หลักการแก้ปัญหาแบบ Divide and Conquer

1. พยายามแบ่งข้อมูลที่มีอยู่ออกเป็นชุดข้อมูล แบบเดิมแต่มีขนาดลดลงหลายๆชุด
2. ทำการแก้ปัญหาในแต่ละชุดย่อย เพื่อให้ได้คำตอบที่ต้องการ
3. นำผลลัพธ์ที่ได้ในแต่ละชุดย่อยมารวมกัน (Merge) จนกระทั่งเป็นข้อมูลชุดเดิมก่อนทำการแบ่ง ซึ่งวิธีการเรียงลำดับข้อมูลที่อาศัยหลักการนี้คือ Quick Sort และ Merge Sort

2.4.2.1 Quick Sort

จะมีความสัมพันธ์ คือ $\{ S_1 \} < X < \{ S_2 \}$ เมื่อ $\{ S_1 \}$คือเซ็ตของข้อมูลที่น้อยกว่า ตัวแบ่ง X_1

$\{ S_2 \}$คือเซ็ตของข้อมูลที่มากกว่า ตัวแบ่ง X_1

มีขั้นตอนการทำงาน

1. เลือกข้อมูลตัวใดตัวหนึ่งเป็นตัวแบ่ง เช่น ใช้ข้อมูลตัวแรกคือ A[1]
2. นำข้อมูลที่มีค่าน้อยกว่าตัวแบ่งมาวางไว้ทางซ้ายของตัวแบ่ง
3. นำข้อมูลที่มีค่ามากกว่าตัวแบ่งมาวางไว้ทางขวาของตัวแบ่ง
4. ทำซ้ำในข้อ 1 ถึง ข้อ 3 กับกลุ่มข้อมูลที่อยู่ทางซ้ายและทางขวาของตัวแบ่ง ในลักษณะเดียวกันจน แบ่งกลุ่มข้อมูลไม่ได้อีกแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. นำข้อมูลทั้งหมดทางด้านซ้าย ด้านขวา และตัวแบ่ง มารวมกันเป็นผลลัพธ์

โปรแกรม Quick Sort สามารถแสดงได้ดังภาพที่ 2.19

```
PROGRAM  quicksort ; (* RECURSIVE CALL PARTITION *)
CONST max = 20 ;
TYPE  itemtype          =     integer;
      arrtype     =     ARRAY [1..max] OF itemtype;
      index       =     0..max;
VAR   a           :     arrtype ;
      L,R,ni,i,K  :     index ;
      num         :     integer;
PROCEDURE  partition(L,R :index);
VAR   i,j     :     index;
      w,x     :     itemtype;
BEGIN
      i     :=    L; j    :=    R;
      x     :=    a[(L+R) DIV 2 ] ;
      REPEAT
            WHILE    a[i] < x  DO  i := i+1;
            WHILE    x < a[j]  DO  j := j-1;
            IF  i <= j  THEN
            BEGIN
                  w     :=    a[i];
                  a[i]  :=    a[k];
                  a[j]  :=    w;
                  i     :=    i+1;
                  j     :=    j-1;
            END;
      UNTIL  i > j;
      IF     L < j  THEN     partition  (L,j);
      IF     I<R   THEN       partition  (I,R);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

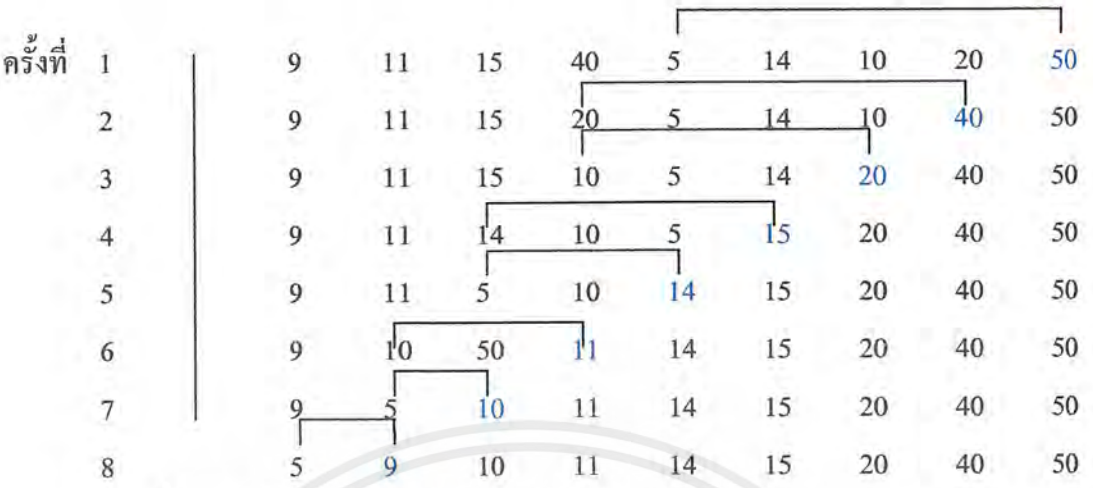
```
END;

BEGIN (*main*)
        WRITELN('ENTER DATA , 999 for end enter');
        n := 0;
        READLN(num);
        WHILE  num<> 999 DO
        BEGIN
                n       :=      n+1;
                a[n]    :=      num;
                READLN(num);
        END;
        Partition (1,n);
        (*write data after sort *)
        WRITELN;
        WRITELN('SORTED DATA');
        FOR  i:= 1 TO  n  DO   WRITE(a[I] :3);
END.
```

ภาพที่ 2.19 โปรแกรม quicksort

ตัวอย่างการทำงานสามารถแสดงได้ด้วยภาพที่ 2.20 แสดงการเรียงลำดับโดยวิธีควิกซอร์ทเป็นกรณีที่แย่ที่ สุดเนื่องจากทุกครั้งที่เลือกค่า X ได้ค่ามากที่สุดซึ่งภาพนี้แสดงชุดข้อมูลที่เปลี่ยนไปในแต่ละครั้งของการแบ่งส่วน ตัวเลขสีน้ำเงินแสดงข้อมูลที่เลือกซึ่งถูก สลับค่ากับข้อมูลตัวสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.20

ประสิทธิภาพของควิคซอร์ท

ความต้องการอันสุดยอดของการแบ่งส่วนข้อมูลคือต้องการให้ค่าที่เลือกไว้นั้นหลังจากแบ่งส่วนแล้วอยู่กลางชุดข้อมูล ทุกๆครั้งที่ทำการแบ่งส่วน ซึ่งในการแบ่งส่วนครั้งแรก จำนวนครั้งของการเปรียบเทียบ ประมาณ n หลังจาก แบ่งส่วนแล้วได้ข้อมูลสองส่วน แต่ละส่วนมี n/2 ข้อมูล ซึ่งมีการเปรียบเทียบ n/2 ครั้งเช่นกัน ต่อไปข้อมูลแบ่งเป็น n/4 จำนวนสี่ส่วน จำนวนครั้งของการเปรียบเทียบของแต่ละส่วนข้อมูลเป็น n/4 แบ่งส่วนข้อมูลไปจนตัวสุดท้ายเหลือ ข้อมูลเพียงตัวเดียว

ดังนั้นจำนวนครั้งการเปรียบเทียบเป็น

$$n + 2(n/2) + 4(n/4) + 8(n/8) + \ldots n(n/n)$$

ซึ่งมี $\log_2 n + 1$ เทอม ดังนั้น จำนวนครั้งการเปรียบเทียบของควิคซอร์ทเป็น $n(\log_2 n + 1)$ หรือมีประสิทธิภาพ เป็น $O(n \log_2 n)$

เมื่อพิจารณากรณีที่แย่ที่สุดคือการแบ่งส่วนคือ ถ้าเลือกข้อมูลที่ใช้ในการเปรียบเทียบ (ค่า X) ที่มีค่ามากที่สุด ข้อมูลถูกแบ่งเป็นสองชุด โดยชุดหนึ่งมีจำนวนข้อมูล n-1 ข้อมูล อีกชุดหนึ่งมีหนึ่งข้อมูล และถ้าการแบ่งส่วนทุกครั้งเลือกได้ข้อมูลที่มีค่ามากที่สุดในชุดข้อมูล จะต้องทำการแบ่งส่วนถึง n-1 ครั้ง จำนวนครั้งการเปรียบเทียบเป็น $n + (n-1) + (n-2) + \ldots 1 = n\frac{(n+1)}{2}$

ซึ่งประสิทธิภาพของควิคซอร์ทในกรณีนี้เป็น $O(n^2)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2.2 Merge Sort

มีขั้นตอนการทำงานดังนี้

1. แบ่งกลุ่มข้อมูลที่ต้องการจัดเรียงออกเป็นกลุ่มข้อมูลจำนวนเท่าๆกัน จนกระทั่งไม่สามารถแบ่งกลุ่มได้
2. ทำการเรียงลำดับข้อมูลในแต่ละกลุ่มด้วยวิธีการ Merge Sort
3. นำผลลัพธ์ที่เรียงลำดับแล้วในแต่ละกลุ่มของข้อมูลมารวมกัน(Merge)จนกระทั่งได้ข้อมูลเดิม

โปรแกรม Merge Sort สามารถแสดงได้ด้วยภาพที่ 2.21 และภาพที่ 2.22 แสดงการเรียงลำดับโดยวิธี Merge Sort ของกลุ่มข้อมูล 4 1 3 6 2 5 8 9

```
MergeSort (Datatype A[] , Int Len)
{       If (Len = = 1)
        Return (A);
        L = Len / 2 ;
        B = & A[ Ø ];
        C = & A[L] ;
        Result1 = MergeSort ( B,L);
        Result2 = MergeSort (C,Len – L);
        Result 3 = Merge (Result 1, Result2 ,Len-L);
        Return (Result3);
}
```

ภาพที่ 2.21 Algorithm ของ MergeSort

ประสิทธิภาพของ Merge Sort

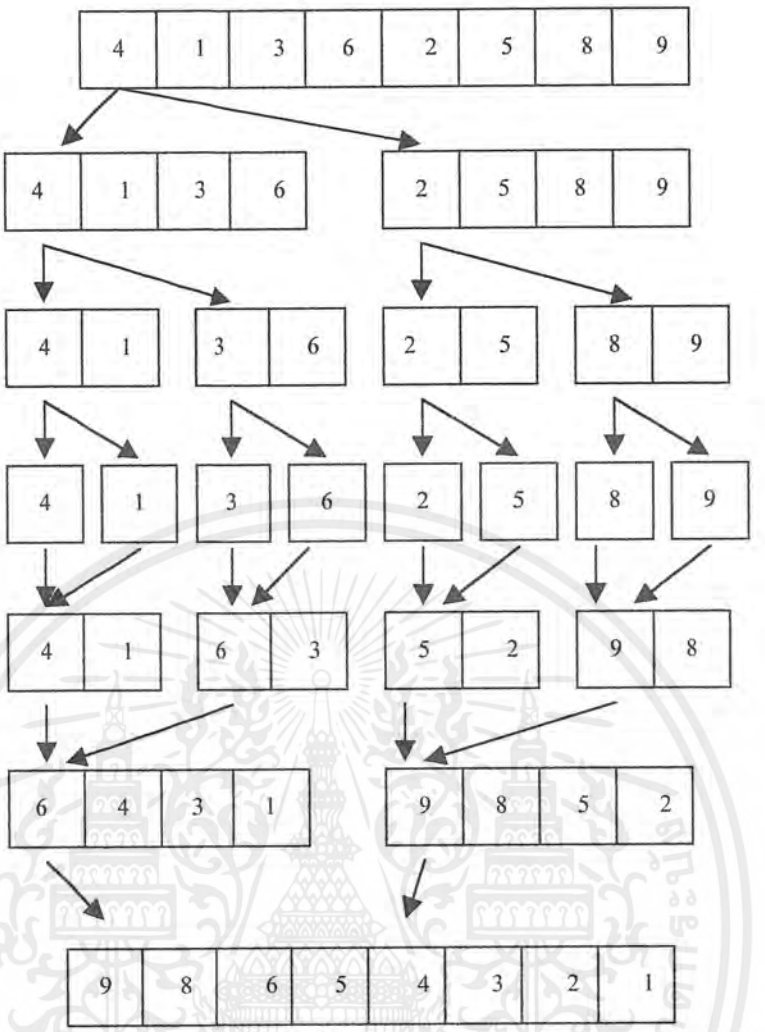
ประสิทธิภาพของ Merge Sort ไม่ขึ้นอยู่กับลักษณะของข้อมูลว่ามีการเรียงลำดับเป็นอย่างไร เนื่องจากขั้นตอนการแบ่งข้อมูล เป็น 2 ส่วน ที่เท่าๆกันนั้น จะไม่สนใจค่าของข้อมูลแต่ละตัว ทำให้เวลาที่ใช้ในการ เรียงลำดับแบบ Merge Sort นี้จะคงที่สำหรับข้อมูลที่มีจำนวนเท่ากัน

จะได้ Time Complexity = $O(n \log_2 n)$ และ Best Case = Average Case = Worst Case

ข้อดี เวลาที่ใช้ในการทำงานไม่ขึ้นอยู่กับลักษณะการจัดเรียงข้อมูล ทำให้ Worst Case คือ $O(n\log_2 n)$ไม่ใช่ $O(n^2)$ เหมือน Quick Sort

ข้อเสีย ใช้ Memory ในการทำงานมาก เนื่องจากจะต้องสร้าง Tree ของข้อมูลเพื่อ Sort โดยทั่วไปจะทำงานช้ากว่า Quick Sort เนื่องจากมีการ Copy ข้อมูลเพื่อเรียงลำดับในแต่ละรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.22 ตัวอย่างการทำงานตามอัลกอริทึมของ Merge Sort

2.4.3 Transposition Sorting

2.4.3.1 Bubble Sort

การเรียงลำดับด้วยการแลกเปลี่ยนเป็นที่รู้จักกันมาก คือ บับเบิลซอร์ท (Bubble Sort) หลักเกณฑ์ของ การเรียงลำดับวิธีนี้คือเปรียบเทียบและแลกเปลี่ยนสมาชิกในลิสต์ที่อยู่ติดกันจนกว่าชุดของข้อมูลในลิสต์จะถูกเรียงลำดับ จำนวนรอบของการเปรียบเทียบและแลกเปลี่ยนจะน้อยลง 1 ทุกรอบไป อัลกอริทึม Bubble Sort เป็นดังนี้

1. ทำข้อ 2-3 สำหรับค่า i=2 ถึง n
2. ทำข้อ 3 สำหรับค่า j= n ลดลงถึง i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าค่า a[j] < a[j-1] (เปรียบเทียบสมาชิกท้าย 2 ตัว) แล้วสลับที่ระหว่าง a[j-1] และ a[j] (สมาชิกใดมี ค่าน้อยกว่าจะลอยขึ้นไป)

4. จบงาน

สามารถเขียนเป็นโปรแกรมปาสคาลได้ดังภาพที่2.23

```
PROCEDURE  Bubblesort (VAR  a:  arrtype  ;  n : index);
VAR   i,j      :        index;
      temp     :        itemtype;
BEGIN FOR   i := 2   TO n   DO
      BEGIN      FOR    j  :=  n  DOWNTO  i    DO
                 IF  a[j]  <  a[j-1]   THEN
                 BEGIN (*exchange*)
                     temp  :=  a[j];
                     a[j]    :=  a[j-1];
                     a[j-1]  :=  temp;
                 END;
      END;
END:
```

ภาพที่ 2.23 Procedure Bubblesort

ภาพที่ 2.24 แสดงการเรียงลำดับตามอัลกอริทึม Bubble Sort โดยเครื่องหมาย_แสดงถึงข้อมูลที่ถูก สับ เปลี่ยนจนขึ้นไปอยู่ที่ต้นของลิสต์

ข้อมูลเดิม	i = 2	i = 3	i = 4	i = 5	i = 6	i=7
5	2	2	2	2	2	2
20	5	5	5	5	5	5
7	20	7	7	7	7	7
10	7	20	10	10	10	10
13	10	10	20	13	13	13
2	13	13	13	20	20	20
30	30	30	30	30	30	30

ภาพที่ 2.24 แสดงการเรียงลำดับตามอัลกอริทึม Bubble Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพของบับเบิลซอร์ทเท่ากับของ Selection Sort คือเท่ากับ (n-1)+ (n-2) +...+1 ได้

$$C = \frac{(n^2 - n)}{2} = O(n^2)$$

2.4.4 Diminishing Increment Sorting

2.4.4.1 Shell Sort

อาศัยหลักการการแบ่งออกเป็นกลุ่มย่อยของกลุ่มข้อมูลทั้งหมดในหลายรอบ โดยแต่ละกลุ่มย่อยจะมีระยะ ห่างคือ h_i; i = 1,2,3,...,t เมื่อมีข้อมูลทั้งหมดคือ n ตัวดังนั้นข้อมูลในแต่ละกลุ่มย่อย จะมีข้อมูลประมาณ $\frac{n}{}$ ข้อมูล จะมีการวนรอบการทำงานจำนวน t รอบ

สรุปว่า Shell Sort จะใช้ $h_1, h_2, h_3, \ldots, h_t$ ซึ่งเรียกว่า Increment Sequence ในการทำงาน

ขั้นตอนการทำงาน

1. Span item แล้วใช้ Insertion Sort
2. Decrement Span หรือลดขนาด Span ลงเพื่อทำงานในรอบต่อไปจนกระทั่ง Span = 1 นั่นคือ h_t

การเลือก Span

1. เลือก Span ที่เป็น Prime Number เช่น 1 , 3 , 5 , 7,...
2. ถ้า Increment Sequence เป็น $h_1, h_2, h_3, \ldots, h_t$ แล้วจะได้ $h_t = 1$, $h_{i+1} < h_i$

การเลือก h_i มีวิธีการคำนวณจากสูตร $h_i = 2^i - 1$ โดยค่า i อยู่ระหว่าง 1 กับ $\log_2 n$

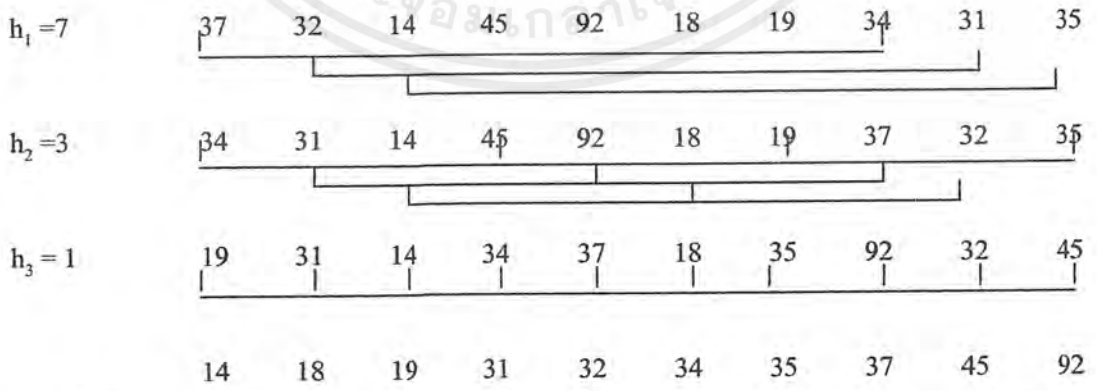
มีข้อมูล n = 10 คือ 37 , 32 , 14 , 45 , 92 , 18 , 19 , 34 , 31 , 35

จาก n = 10 ดังนั้น $[\log_2 10] = 3$ ดังนั้น i = 1 , 2 , 3

ดังนั้น i = 1 , $h_1 = 2^1 - 1 = 1$

i = 2 , $h_2 = 2^2 - 1 = 3$

i = 3 , $h_3 = 2^3 - 1 = 7$



ภาพที่ 2.25 แสดงตัวอย่างการหาค่า Increment Sequence และการเรียงลำดับโดย Shell Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 Address Calculation Sorting

2.4.5.1 Proxmap Sort

มีขั้นตอนการทำงานดังนี้

1. พิจารณาข้อมูลแต่ละข้อมูลเรียก Key แทนด้วย K ใน Array A
2. เลือกใช้ฟังก์ชั่น Map Key ตัวอย่างเช่น Mapkey (K) = Seiling(K) เลือก Seiling หรือ Floor
3. หา Hit Count

```
FOR i := 1 TO n DO
BEGIN
        j   := MapKey (A[I]);
        H[j] := H[j] +1 ;
END;
```

4. หา Proxmap

```
Running total := 1
FOR i := 1 TO n DO
BEGIN
    IF H[i] > Ø then
    BEGIN
        P[i] := Runningtotal + H[i];
    END;
END;
```

5. หา Future Insertion Location

```
FOR I:= 1 TO N DO
BEGIN
 L[i] := P[Mapkey (A[i])];
END;
```

6. Moving Keys.

การย้าย A[i] ไปยังตำแหน่ง Array L[i]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5.2 Radix Sort

มีขั้นตอนในการทำงานดังต่อไปนี้

1. จะใช้แต่ละตำแหน่งของข้อมูลเป็นคีย์ในการพิจารณาโดยเริ่มตั้งแต่ตำแหน่งนัยสำคัญน้อยที่สุด (Least Significant Digit)
2. ใช้ถังสำหรับเก็บข้อมูล (อาจเป็น Queue หรือ Array ก็ได้)
3. ในการทำงานแต่ละรอบจะพิจารณาค่าของ Key แล้วนำข้อมูลไปใส่ถัง หลังจากนั้นให้ดึงข้อมูลจากถัง (โดยเรียงลำดับจากถังแรกไปยังถังสุดท้ายและในถังเดียวกัน ให้ดึงข้อมูลโดยอาศัยหลักการ FIFO)

เรียงข้อมูล 156 , 425 , 679 , 128 ,420 , 067 , 772 , 895 , 835 จะต้องใช้ถังจำนวน 10 ถัง ในการเรียงลำดับ ตัวเลขแสดงการทำงานได้ดังภาพที่ 2.26

	รอบที่ 1	รอบที่ 2	รอบที่ 3
ถังที่0	420		067
ถังที่1			128,156
ถังที่2	772	420,425,128	
ถังที่3		835	
ถังที่4			420,425
ถังที่5	425,895,835	156	
ถังที่6	156	067	679
ถังที่7	067	772,679	772
ถังที่8	128		835,895
ถังที่9	679	895	

รอบที่ 1 ได้ผลลัพธ์ 420 , 772 , 425 , 895 , 835 , 156 , 067 , 128 , 679

รอบที่ 2 ได้ผลลัพธ์ 420 , 425 , 128 , 835 , 156 , 067 , 772 , 679 , 895

รอบที่ 3 ได้ผลลัพธ์ 067 , 128 , 156 , 420 , 425 , 679 , 772 , 835 , 895

ภาพที่ 2.26 แสดงตัวอย่างการเรียงข้อมูลโดยใช้ Radix Sort

2.4.6 Insert and keep Sorted

2.4.6.1 Insertion Sort

การเรียงลำดับด้วยการแทรกนี้ เป็นวิธีการที่แทบทุกคนเคยใช้เป็นวิธีการที่ง่ายไม่ซับซ้อน เช่นในขณะที่เล่น ไพ่เรามักจะจัดเรียงไพ่ตามเลขโดยการหยิบมาแทรก เช่นเลขชุดหนึ่งเป็น 5 20 7 10 13 2 30

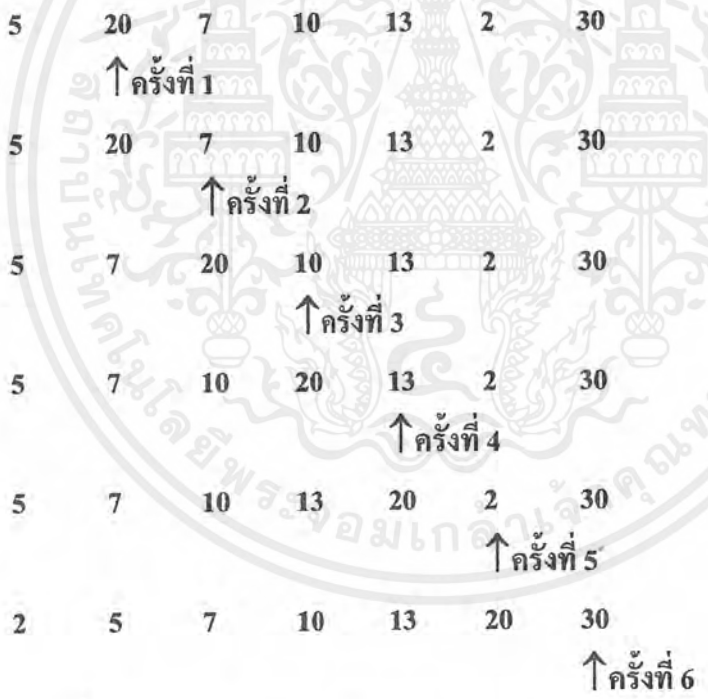
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีจำนวนหมายเลขอยู่ 7 (n=7) วิธีการเรียงลำดับด้วยการแทรกมีขั้นตอนดังนี้ โดยที่ข้อมูลชุดนี้แทนที่ในอาร์เรย์ a ที่เป็น ARRAY[0..n] OF itemtype

อัลกอริธึมการเรียงลำดับแบบการแทรก เป็นดังนี้

ทำตั้งแต่ข้อมูลตัวที่สองถึงตัวสุดท้ายนำมาแทรกในตำแหน่งที่ทำให้ข้อมูลเรียงลำดับโดยขั้นตอนการนำมาแทรกเป็นดังนี้ จะนำค่าที่หยิบให้ชื่อว่า x แต่ละครั้งมาเปรียบเทียบ ใช้ตัวแปร j ชี้สมาชิกที่เปรียบเทียบ กับค่าข้างหน้า x นั้นไปเรื่อย ถ้า x มีค่าน้อยกว่า a[j] จะให้ค่า a[j] มาแทนที่ค่าในตำแหน่ง a[j+1] คือเลื่อนค่า ที่อยู่ข้างหน้าถอยมาอีกหนึ่งตำแหน่ง แล้วเลื่อนไปเปรียบกับตัวด้านหน้าถัดไป (คือ j ถูกลดลง 1) ทำเช่นนี้ไปเรื่อยๆ จนถึงสมาชิกตัวแรก หรือเมื่อพบว่า x มีค่ามากกว่าหรือเท่ากับค่า a[j] ก็จะจบขบวนการนี้

เมื่อเขียนโปรแกรมจะใช้เทคนิคการใช้เซนทิเนลช่วย คือ ให้อาร์เรย์ตำแหน่งที่ศูนย์เท่ากับค่า x (a[0] := x) และเพื่อลดจำนวนคำสั่งลง จึงใช้การเลื่อน a[j] ไปอยู่ที่ a[j+1] แทนที่จะใช้การสับเปลี่ยนค่าเมื่อพบว่า x < a[j]



ภาพที่ 2.27 แสดงตัวอย่างการเรียงข้อมูลด้วยการแทรก

ภาพที่ 2.27แสดงขั้นตอนการเรียงลำดับด้วยการแทรก จะมีการหยิบค่า x ทั้งหมด 6 ครั้ง ในที่นี้เมื่อหยิบ 20 เมื่อมีการเปรียบเทียบจะไม่มีการเลื่อนเปลี่ยนค่าใดๆ การหยิบครั้งที่สองได้ 7 ค่า 20 จะเลื่อนมาณ ตำแหน่งที่ 7 อยู่เดิม การหยิบครั้งที่สามได้ 10 ค่า 20 จะเลื่อนมาแทนที่ค่า 10 เพื่อให้ 10 ไปแทรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทน การหยิบครั้งที่สี่ทำให้ 20 เลื่อนลงมา การหยิบครั้งที่ห้า ตัวที่อยู่ข้างหน้า 2 ทุกตัวถูกเลื่อนลงมาหมดตั้งแต่ 20, 13, 10, 7 และ 5 โดยที่ 2 ไปแทนที่ 5 การหยิบครั้งที่ 6 ไม่มีการเลื่อนค่าลงเลย เมื่อหยิบจนหมดสมาชิกในอาร์เรย์ เป็นอันว่าสิ้นสุดการเรียงลำดับ

อัลกอริธึมของการเรียงลำดับแบบแทรกที่เป็นสัญลักษณ์มากขึ้นเป็นดังนี้

1. ทำข้อ 2 ถึงข้อ 3 สำหรับ i = 2, 3, . . . , n (i <= n)
2. X ← a[i] (X เป็นตัวแปรเก็บค่า a[i] ชั่วคราว)
3. (หาตำแหน่งในอาร์เรย์ที่แทรกค่า X ไว้ เพื่อให้ข้อมูลในอาร์เรย์เรียงลำดับ)
   - 3.1 a[0] ← X (เป็นการใช้ a[0] เป็นเซนทิเนล)
   - 3.2 j ← i − 1
   - 3.3 ในระหว่างที่ X < a[j] ทำ
     (แทนค่า a[j+1] ด้วย a[j] และลดค่า j อีก 1 คือสมาชิกในอาร์เรย์ด้านหน้าเปรียบเทียบกับค่า X อีก)
     - 3.3.1 a[j+1] ← a[j]
     - 3.3.2 j ← j − 1
   - 3.4 (เมื่อพบ X >= a[j])
     a[j+1] ← X
4. จบงาน

ในกระบวนการข้อ 3 (แทรกข้อมูล X ไว้ในอาร์เรย์ในตำแหน่งที่ทำให้อาร์เรย์ตั้งแต่ a[1]..a[i] เรียงลำดับ) ทำการเปรียบเทียบ X กับค่า a[j] โดย j เป็นตำแหน่งค่าในอาร์เรย์ที่อยู่ข้างหน้าของค่า a[i] ค่า a[j+1] จะถูกแทนที่ด้วย a[j] หรือค่า X ขึ้นอยู่กับผลการเปรียบเทียบค่า X กับ a[j] ถ้า X < a[j] (เงื่อนไขในข้อ 3.3 เป็นจริง) ค่า a[j+1] ก็จะถูกแทนที่ด้วย a[j] ถ้า X = a[j] (เงื่อนไขข้อ 3.3 เป็นเท็จ) ค่า a[j+1] ถูกแทนที่ด้วย X

จากอัลกอริธึมข้างต้นสามารถเขียนเป็นโพรซีเซอร์ insertionsort (a, n) และวิธีการเรียงลำดับแบบนี้เรียกว่า insertionsort ภาพที่ 2.28 เป็นโปรแกรม sort ที่มีโพรซีเซอร์ insertion (a, n) เป็นโปรแกรมย่อย

```
PROGRAM sort;
USES   crt;
CONST          n = 10;
TYPE   itemtype : integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        index = 0..n;
        arrtype = ARRAY [0..N] OF integer;
VAR     a : arrtype;
        data, count : index;
PROCEDURE insertionsort (VAR a : arrtype; n : index);
VAR     i, j : index;
        x : itemtype;
BEGIN
        FOR i := 2 TO N DO
        BEGIN x := a[i];
                a[0] := x;
                j := i-1;
                WHILE x < a[j] DO
                BEGIN
                        a[j+1] := a[j];
                        j := j-1;
                END;
        a[j+1] := x;
        END;
END;
BEGIN (*main*)
        writeln('ENTER DATA STOP BY 0       ');
        count := o;
        READLN(data);
        WHILE data <> 0 DO
        BEGIN
                count := count+1;
                a[count] := data;
                READLN(data);
        END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
insetrionsort (a, count);
END.
```

ภาพที่ 2.28 โปรแกรม sort (แสดงโพรซีเซอร์ insertionsort (a, n))

ภาพที่ 2.29 แสดงการทำงานของโพรซีเซอร์ insertionsort (a, n) โดยมีค่า i (เริ่มจาก 2 ถึง n) กำกับในแต่ละขั้นตอน ค่าใน a[0] ใช้เป็นเซนทิเนลจะเปลี่ยนไปทุกๆ รอบ I

	ดัชนี	0	1	2	3	4	5	6	7
	ข้อมูลเดิม		5	20	7	10	13	2	30
i = 2	(*) j = 1	20	5	20	7	10	13	2	30
i = 3	j = 2	7	5	20	20	10	13	2	30
	(*) j = 1	7	5	7	20	10	13	2	30
i = 4	j = 3	10	5	7	20	20	13	2	30
	(*) j = 2	10	5	7	10	20	13	2	30
i = 5	j = 4	13	5	7	10	20	20	2	30
	(*) j = 3	13	5	7	10	13	20	2	30
i = 6	j = 5	2	5	7	10	13	20	20	30
	j = 4	2	5	7	10	13	13	20	30
	j = 3	2	5	7	10	10	13	20	30
	j = 2	2	5	7	7	10	13	20	30
	j = 1	2	5	5	7	10	13	20	30
	(*) j = 0	2	2	5	7	10	13	20	30
i = 7	j = 6	30	2	5	7	10	13	20	30

ภาพที่ 2.29

ประสิทธิภาพของ insertionsort

จากโพรซีเซอร์ insertionsort (a, n) มาพิจารณาจำนวนครั้งการเปรียบเทียบ (C) ในแต่ละรอบ i (2 ถึง n) มีการเปรียบเทียบน้อยที่สุดคือ 1 ครั้ง เช่น เมื่อ i = 2 และเมื่อ i = 7 การเปรียบเทียบมากที่สุดในแต่ละรอบ i เป็น i ครั้ง เช่นเมื่อ i = 6 ต้องเปรียบเทียบค่า a[j] กับ X ตั้งแต่ j = 5, 4, 3, 2, 1 จนถึง 0 จึงพบว่าเงื่อนไขใน WHILE...DO เป็นเท็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้ C_{min} เป็นจำนวนครั้งการเปรียบเทียบน้อยที่สุด

C_{max} เป็นจำนวนครั้งการเปรียบเทียบมากที่สุด

C_{avg} เป็นจำนวนครั้งการเปรียบเทียบเฉลี่ยแล้ว

ดังนั้น $C_{min} = n-1 = O(n)$

$C_{max} = \frac{n(n+1)}{2}-1 = \frac{n^2+n-2}{2} = O(n^2)$

$C_{avg} = \frac{1}{4}(n^2+n-2) = O(n^2)$

ให้สังเกตว่าในแต่ละครั้งที่นำ a[i] ไปแทรกเข้าระหว่าง a[1]..a[i-1] นั้น ค่าใน a[1]..a[i-1] เรียงลำดับอยู่แล้ว เราสามารถปรับปรุงอัลกอริธึม insertionsort ให้ทำงานมีประสิทธิภาพขึ้นโดยการค้นหาตำแหน่งที่จะแทรก a[i] โดยการค้นหา a[1]..a[i-1] อย่างทวิภาค (Binary Search) ซึ่งทำการแยกลิสต์ของข้อมูลที่ต้องการค้นหาเป็น 2 ส่วนทุกครั้งไปถึงอย่างไรก็ตามถ้าข้อมูลที่นำมาเรียงลำดับเป็นข้อมูลที่ค่อนข้างจะเรียงลำดับอยู่แล้วการใช้วิธีเรียง ลำดับโดยใช้วิธีการค้นหาแบบทวิภาคเข้าช่วย จะทำงานช้ากว่าแบบปกติ จึงมีข้อสรุปอยู่ข้อหนึ่งว่า การเรียงลำดับ แบบการแทรกนี้เหมาะกับข้อมูลที่ค่อนข้างเรียงลำดับอยู่แล้ว ลองมาเรียงลำดับข้อมูลชุดนี้ 1 4 8 10 14 12 20 ดูว่า ถ้าใช้ขั้นตอนตามอัลกอริธึม insertionsort จะเหมาะสมอย่างไร

	ดัชนี	0	1	2	3	4	5	6	7
	ข้อมูลเดิม		1	4	8	10	14	12	20
i = 2	(*) j = 1	4	1	<u>4</u>	8	10	14	12	20
i = 3	(*) j = 2	8	1	4	<u>8</u>	10	14	12	20
i = 4	(*) j = 3	10	1	4	8	<u>10</u>	14	12	20
i = 5	(*) j = 4	14	1	4	8	10	<u>14</u>	12	20
i = 6	j = 5	12	1	4	8	10	14	14	20
	(*) j = 4	12	1	4	8	<u>10</u>	12	14	20
i = 7	(*) j = 6	20	1	4	8	10	12	14	<u>20</u>

ภาพที่ 2.30

เห็นว่าในแต่ละรอบของ I มีการเปรียบเทียบเพียง 1 ครั้ง ยกเว้นในรอบที่ I = 6 มีการเปรียบเทียบ 2 ครั้ง ดังนั้นในการเรียงลำดับข้อมูลที่ค่อนข้างเรียงลำดับอยู่แล้วจะมีการเปรียบเทียบใกล้เคียง C_{min} มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.6.2 Tree Sort

การเรียงลำดับแบบทรีซอร์ทนี้อาศัยต้นไม้ไบนารีบรรจุข้อมูลแทนลิสต์โดยมีขั้นตอน 2 ขั้นตอน

1. สร้างต้นไม้ไบนารีจากชุดข้อมูลซึ่งข้อมูลที่น้อยที่สุดจะอยู่ที่รู้ตของต้นไม้
2. จากนั้นนำค่าที่รู้ตของต้นไม้เป็นผลลัพธ์ของการเรียงลำดับและทำการเลือกค่าที่น้อยที่สุดของข้อมูลที่เหลือให้ขึ้นไปที่รู้ต

ขั้นตอนแรกทำเพียงครั้งเดียวในครั้งแรก หลังจากนั้นทำขั้นตอนที่สองจนข้อมูลในต้นไม้หมดผลจะได้ ข้อมูลที่เรียงลำดับ เช่น ข้อมูลชุดนี้ (n = 8) 12 10 4 6 9 15 30 22 ดังภาพที่ 2.31 เป็นต้นไม้ ที่สร้างขึ้นโดยการเลือกข้อมูลที่น้อยในแต่ละคู่ ขึ้นไปอยู่ในระดับต่อไป ดังนี้

ครั้งที่ 1	12, 10	เลือก	10
	4, 6	เลือก	4
	9, 15	เลือก	9
	30, 22	เลือก	22

เห็นว่ามีจำนวนครั้งการเปรียบเทียบเป็น n/2 และในครั้งต่อๆ ไปมีจำนวนการเปรียบเทียบ n/4 และ n/8 ดังนี้

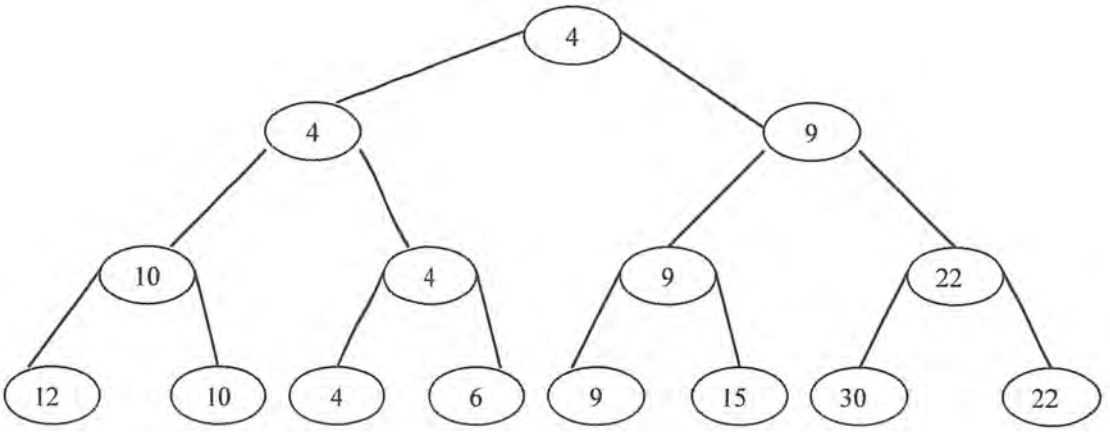
ครั้งที่ 2	10, 4	เลือก	4
	9, 22	เลือก	9
ครั้งที่ 3	4, 9	เลือก	4

จากจำนวนการเปรียบเทียบในแต่ละครั้ง เมื่อรวมกันเป็น $n/2 + n/4 + n/8 + \ldots + 1 = n - 1$

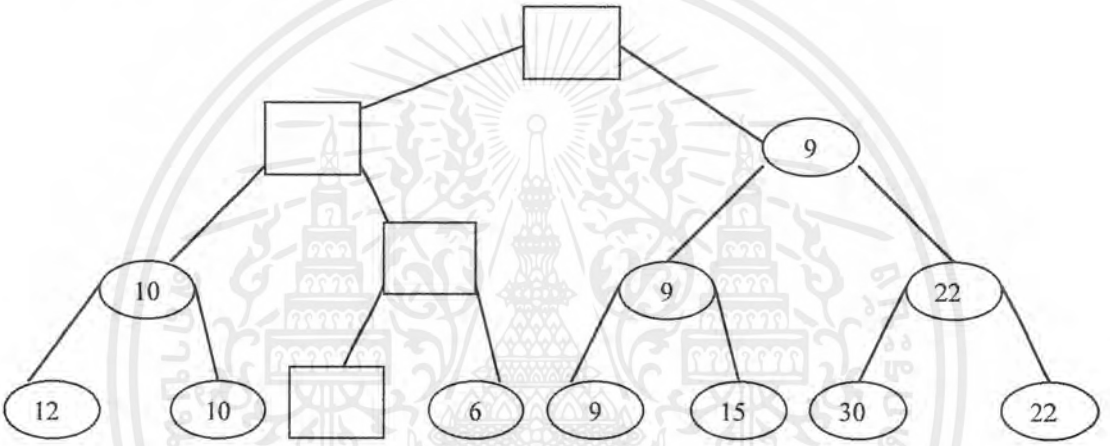
ในขั้นตอนที่สองของทรีซอร์ท คือ นำค่าในรู้ตออกเป็นผลลัพธ์และแทนโหนดที่มีข้อมูลที่นำออกด้วยค่าที่ มากสุดเพื่อค่านั้นจะไม่ได้ถูกเลือก แต่ขอแสดงด้วยรูปสี่เหลี่ยม (□) แทนดังแสดงในรูปที่ 2.32 หลังจากนั้นเลือกข้อมูล ที่มีค่าน้อยที่สุดของข้อมูลที่เหลือขึ้นมาแทนรู้ตจะได้ 6 ซึ่งเลข 6 ขึ้นไปตามเส้นทางที่แสดงด้วยลูกศรซึ่งก็เกิดจาก การเปรียบเทียบค่า ระหว่างคู่เหมือนเดิมคือ (□) กับ 6 เลือก 6 และ 10 กับ 6 เลือก 6 และท้ายสุด 6 กับ 9 เลือก 6 ซึ่งจำนวนครั้งการเปรียบ เทียบนี้ไม่เกินความสูงของต้นไม้คือ $\log_2 n$ และทำเช่นนี้ n ครั้งต้นไม้นี้จะมีแต่ (□) ซึ่งหมายถึงข้อมูลได้เรียงลำดับเรียบร้อยแล้ว และจำนวนครั้งการเปรียบเทียบทั้งหมดเป็น $n\log_2 n$ หมายถึงทรีซอร์ท มีประสิทธิภาพเป็น $O(n\log_2 n)$ ซึ่งดีกว่าเซเล็กชันซอร์ทที่เป็น $O(n^2)$

อย่างไรก็ตาม ทรีซอร์ท ต้องใช้เนื้อที่เก็บมากเกินความจำเป็น (ภาพที่ 2.31) ต้องใช้เนื้อที่ 2n-1 และมี จำนวนการเปรียบเทียบที่ไม่จำเป็น เช่นเมื่อเปรียบเทียบข้อมูลกับค่าสูงสุดที่แสดงด้วย ดังนั้น จึงมีการกำหนดต้นไม้ พิเศษขึ้นเรียกว่า ต้นไม้ฮีป เพื่อใช้ในการเรียงลำดับด้วยวิธี ฮีปซอร์ท ซึ่งเนื้อที่ทีใช้เป็น n เนื้อที่เท่านั้น

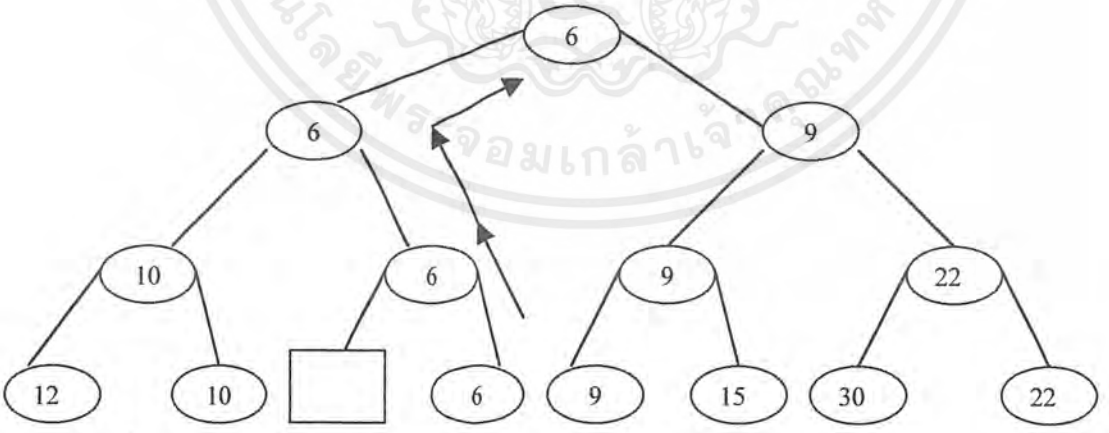
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.31 ทรีของข้อมูล 12 10 4 6 9 15 30 22



ภาพที่ 2.32 เมื่อดึงข้อมูล 4 ออกจากทรีและแทนค่าด้วยค่าที่มากๆ



ภาพที่ 2.33 ทำการPromote ค่า 6 ซึ่งมีค่าน้อยที่สุดขึ้นมาไว้ที่ Root Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.7 Comparison and Perspective

1. เพื่อที่จะเปรียบเทียบประสิทธิภาพในการทำงานในการเรียงลำดับข้อมูลด้วยวิธีการต่างๆ
2. เพื่อผู้ใช้สามารถเลือกใช้สามารถเลือกใช้วิธีการเรียงลำดับข้อมูลได้อย่างเหมาะสม

ตารางที่ 2.5 แสดงผลของเวลาที่ใช้ในการจัดเรียงข้อมูลด้วยวิธีการในการจัดเรียงข้อมูลต่างๆ

Array Size	N=64	N=128	N=256	N=512	N=1024
Quick Sort	0.40	0.98	2.22	4.94	10.86
Heap Sort	0.61	1.43	3.28	7.43	16.57
Proxmap Sort	0.38	0.75	1.51	3.00	5.99
Shell Sort	0.42	1.04	2.37	5.44	11.97
Bubble Sort	2.76	11.36	46.42	189.35	766.22
Insertion Sort	1.12	4.47	17.58	69.89	280.27
Selection Sort	1.40	5.56	22.18	88.66	254.48
Merge Sort	0.99	2.28	5.13	11.45	25.11

จากตารางที่ 2.5ซึ่งแสดงผลของเวลาที่ใช้ในการจัดเรียงข้อมูลด้วยวิธีการในการจัดเรียงข้อมูลต่างๆ โดยมีขนาด ของอาเรย์ที่แตกต่างกัน 5 ขนาด ได้แก่ 64 , 128 , 256 , 512 , 1024 ซึ่งอารเรย์ A เก็บข้อมูลตัวเลขทศนิยม ที่ได้จากการสุ่มแบบ Uniform โดยมีค่าอยู่ในช่วง 10<= r <1 โดยหน่วยที่ใช้ในการวัดคือ Millitick (1 Millitick เท่ากับ 1/60000 วินาที) โดยค่าเฉลี่ยที่ได้เป็น ค่าเฉลี่ยจากการรันโปรแกรมแต่ละอัลกอลิทึมจำนวน 100 ครั้ง

จากการตารางที่ 2.5 พบว่าถึงแม้ Mapkey Operation จะเสียเวลามากกว่าเป็น 10 / 3 เท่าของการ Comparing ค่า 2 ค่า แต่ Proxmap Sort ก็ยังทิ้งห่างเอาชนะคู่แข่งคือ Quick Sort ด้วยอัตราส่วน 1.8 สำหรับ อาเรย์ที่มีข้อมูล 1024 คีย์ Shell Sort และ Heap Sort ก็เป็นเทคนิคการ Sort ที่ดีที่ใช้ภายใต้เงื่อนไขนี้ด้วยเช่นกัน

เทคนิคที่มีประสิทธิภาพ $O(n^2)$ เป็นเทคนิคที่แย่ที่สุดท่ามกลางเทคนิคต่างๆที่ถูกเปรียบเทียบในตารางที่ 2.5 ซึ่ง Insertion Sort เอาชนะ selection Sort ด้วยอัตราส่วน 1.26 และเอาชนะ Bubble Sort ด้วยอัตราส่วน 2.73 ท่ามกลาง Comparison Based Method วิธีที่ดีที่สุดด้วยขนาดของอาเรย์ 1024 คือ Quick Sort ซึ่งเอาชนะเทคนิคที่แย่ที่สุดคือ Bubble Sort ด้วยอัตราส่วน 70 ด้วยขนาดของอาเรย์ 1024

ถึงแม้ Heap Sort ซึ่งมี Average Running Time เป็น 2เท่าของ Quick Sort แต่ Heap Sort ก็ยังเป็นทางเลือกที่เหมาะสมและดีกว่าสำหรับการจัดเรียงข้อมูลที่ต้องการเวลาที่มีลักษณะเป็น Real Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แม้นว่า Proxmap Sort เป็นเทคนิคที่แย่ที่สุดภายใต้ข้อกำหนดของการทดลองนี้แต่ก็มีบางครั้งที่การทำงาน ของ Sort อัลกอลิทึมนี้ไม่เหมาะสมนั่นคือสถานการณ์ที่ข้อมูลในอาร์เรย์ A ที่จะทำการเรียงลำดับนั้นรวมกันอยู่เป็นกลุ่ม ในช่วงหนึ่งของ Key Space เช่นการเรียงลำดับชื่อในสมุดโทรศัพท์โดยรายชื่อนั้นขึ้นต้นด้วยตัวอักษร M หรือเมื่ออยู่ในสถานการณ์ที่ค่าใช้จ่ายในการคำนวณ Mapkey Function นั้นแพงกว่าเมื่อเทียบกับค่าใช้จ่ายที่เกิดจากการ เปรียบเทียบและเคลื่อนย้ายคีย์

ข้อเสียหนึ่งของ Proxmap Sort คือเรื่องพื้นที่ที่ใช้ในการเก็บข้อมูลโดย Proxmap Sort ต้องการเนื้อที่ พิเศษ สำหรับบิตสถานะที่เพิ่มขึ้นมา 2 บิตในทุกๆเรคคอร์ดของทุกๆ Entry ในอาร์เรย์ A ซึ่งถ้าพื้นที่พิเศษไม่เพียงพอ ประสิทธิภาพของ Proxmap Sort ก็จะแย่ลงด้วย

หากพื้นที่หน่วยความจำไม่เพียงพอ Quick Sort ก็ไม่ใช่เทคนิคที่ดีที่สุดสำหรับเทคนิคที่มีประสิทธิภาพ O(nlog n) เนื่องจาก Quick Sort นั้นใช้พื้นที่ในการเก็บ Stack เพื่อเรียกใช้ตัวเองดังนั้น Heap Sort น่าจะเป็นทางเลือกที่ดีกว่าในสถานการณ์เช่นนี้

2.5 Active Server Pages

2.5.1 Active Server Pages คืออะไร

Active Server Pages เป็นเอกสาร HTML ที่ได้เพิ่มความสามารถบางอย่างเข้าไป ดังนั้น Active Server Pages จึงสามารถทำงานหลายๆ สิ่งที่ HTML ทำได้เช่น Active Server Pages สามารถบรรจุ HTML tags ซึ่งจะถูกแปล (Interpret) และแสดงผลโดย Browser และ อีกหลายๆ สิ่งที่เราใช้บรรจุให้กับหน้าเอกสาร HTML ธรรมดาเช่น Java applets, ตัวอักษรกระพริบ, Script ที่ใช้ประมวลผลทาง Client (Browser), ActiveX control ทาง Client เหล่านี้สามารถนำมาไว้ที่ Active Server Pages ได้ แต่อย่างไรก็ตาม Active Server Pages ยังมีคุณลักษณะ (features) ที่สำคัญ 4 อย่างที่ทำให้โดดเด่นนั้น คือ

1. Active Server Pages สามารถบรรจุ Script ที่ใช้ประมวลผลทาง Server ได้การทำเช่นนี้ได้ทำให้เกิด ประโยชน์มากเพราะจะทำให้เราสามารถสร้างหน้าเอกสารที่เป็น dynamic ได้ตัวอย่างง่ายๆที่แสดงถึงประโยชน์ของ คุณลักษณะนี้คือเราสามารถที่จะสร้างเอกสาร(page)ที่แสดงข้อความทักทายที่แตกต่างกันไปในแต่ละเวลาของวันได้

2. Active Server Pages ได้เตรียม built-in object มากมาย การที่มี built-in object ใน Active Server Pages ช่วยให้ Script ของเรามีประสิทธิภาพในการใช้งานมากขึ้น นั้นเพราะ Object ต่างๆ เหล่านี้จะทำให้เราสามารถรับ-ส่ง ข้อมูล (data) ระหว่าง Server กับ Client (Browser) ได้ ตัวอย่างเช่น การใช้ Object "Request" เราสามารถรับข้อมูลจาก ผู้ใช้ที่ส่ง มา ทางฟอร์ม (Form) ของ HTML และส่งข้อมูลนั้นต่อไปให้กับส่วนของ Script ที่ต้องการ ได้อย่างง่ายดาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Active Server Pages สามารถเพิ่ม Component ที่ต้องการได้ ไม่เฉพาะ Component มาตรฐานที่ Active Server Pages ได้เตรียมไว้ตอนที่ Install เท่านั้น Active Server Pages ยังสามารถทำการเพิ่ม Component ที่ต้องการ เข้าไปได้อีก

4. Active Server Pages สามารถทำการติดต่อกับฐานข้อมูล (Database) ดังเช่น Microsoft SQL Server หรือ Microsoft Access ได้เป็นอย่างดี โดยการใช้ชุดของ Object พิเศษ (Object เหล่านี้มีเป็นมาตรฐานอยู่แล้วใน Active Server Pages) ที่เรียกว่า ActiveX Data Object (ADO) คุณลักษณะในข้อนี้ทำให้ Active Server Pages มีประสิทธิภาพมาก ที่สุด ในการที่จะนำไปใช้งาน

ดังนี้ด้วยคุณลักษณะทั้ง 4 ข้อที่กล่าวมา ทำให้กล่าวได้ว่า Active Server Pages นั้นคือหน้าเอกสาร HTML (pages) มาตรฐานที่ได้เพิ่มการทำงานของ Script ที่ประมวลผลทาง Server โดยมี Object และ Component เพิ่มเข้า มาช่วยในการทำงาน ทำให้สามารถสร้าง Web Site ที่มี หน้าเอกสาร (pages) แบบ Dynamic ได้

ในการใช้งาน Active Server Pages นี้เราสามารถใช้กับ Web Server ที่มาจาก Microsoft ดังเช่น Internet Information Server (IIS) สำหรับ Window NT Server, Microsoft's Personal Web Server สำหรับ Window 95/98, Peer Web Server สำหรับ WindowNT Workstation หรือ ของบริษัทอื่นๆ ที่ไม่ใช่ Microsoft เช่น Chili!ASP ของ Chili!Soft, FastTrack servers, LotusDomino, Go Servers และอื่นๆ อีกมากมาย

2.5.2 Active Server Pages ทำอะไรได้บ้าง

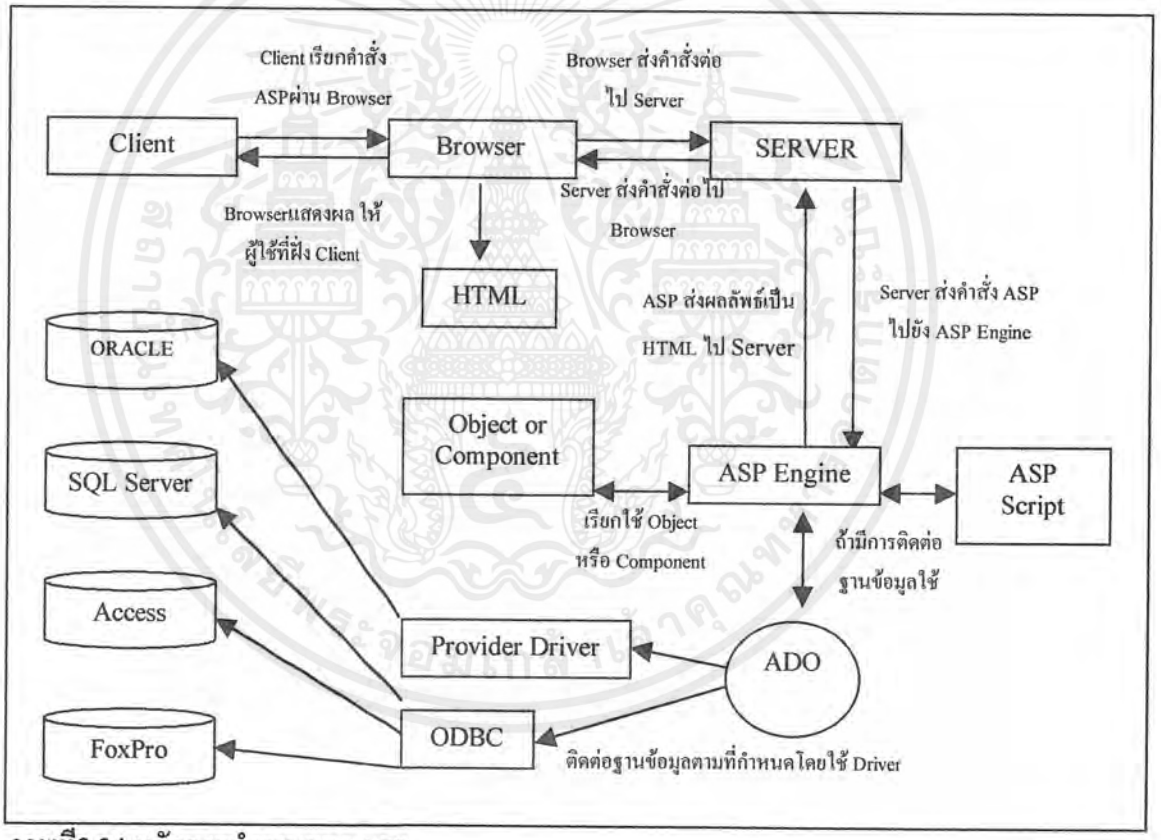
Active Server Pages สามารถทำประโยชน์ได้อย่างไม่มีขีดจำกัดรายการด้านล่างนี้ จะแสดงถึงตัวอย่างบาง ส่วนที่ Active Server Pages สามารถทำได้

1. รับข้อมูลจากฟอร์ม HTML และเก็บข้อมูลนั้นที่ฐานข้อมูล (Database)
2. สร้างหน้าเอกสาร (Pages) ส่วนบุคคลที่จะแสดงรายละเอียดแตกต่างกัน เป็นส่วนเฉพาะของแต่ละบุคคล
3. แสดงหน้าเอกสาร (Pages) ที่แตกต่างกันสำหรับความสามารถที่มีของแต่ละ Browser ของผู้ใช้
4. ทำตัวเชื่อมโยง (Link) ร่วมกันของหลายๆ หน้าเอกสาร ทำให้เกิดความง่ายในการค้นหา
5. ทำตัวนับ (Counter) ให้กับหน้าเอกสาร (Pages) มากกว่าหนึ่งหน้า
6. เก็บข้อมูลเกี่ยวกับกิจกรรมที่ผู้ใช้ทำต่อ Web Site ลงบน log ไฟล์ได้ ดังนี้ เป็นเพียงตัวอย่างง่ายๆ ที่ Active Server Pages สามารถทำได้อาจกล่าวได้ว่าสิ่งต่างๆที่ผู้พัฒนา Web Site ต้องการสามารถหาอุปกรณ์เหล่านั้นได้ จากActive Server Pages เลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 Active Server Pages ทำงานอย่างไร

เนื่องจาก ASP จะทำงานโดยมีตัวแปลและเอ็กซิคิวต์ที่ฝั่งเซิร์ฟเวอร์ อาจจะเรียกการทำงานว่าเป็นเซิร์ฟเวอร์ไซด์ (Server Side) ส่วนการทำงานของบราวเซอร์ของผู้ใช้เรียกว่าไคลเอ็นต์ไซด์ (Client Side) โดยการทำงานจะเริ่มต้นที่ผู้ใช้ส่งข้อความต้องการผ่านเว็บบราวเซอร์ทาง HTTP (HTTP Request) ซึ่งอาจจะเป็นการกรอกแบบฟอร์ม หรือใส่ข้อมูลที่ต้องการ ข้อมูลเหล่านั้นจะเป็นเอกสาร ASP (เอกสารนี้จะมีส่วนขยายเป็น asp เช่น Search.asp เป็นต้น) เมื่อเอกสาร ASP เข้ามาถึงเว็บเซิร์ฟเวอร์ก็จะถูกส่งไปให้ ASP เพื่อทำหน้าที่แปลคำสั่งและเอ็กซิคิวต์คำสั่งนั้น ซึ่ง ASP อาจจะเรียกใช้ออปเจ็กต์ , คอมโพเนนต์หรือ ADO (เพื่อใช้ฐานข้อมูล)หลังจากนั้น ASP จะสร้างผลลัพธ์ในรูปแบบเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปให้บราวเซอร์แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP Response) ซึ่งลักษณะการทำงานแบบนี้จะคล้ายกับการทำงานของ CGI (Common Gateway Interface) หรืออาจกล่าวได้ว่า ASP ก็คือโปรแกรม CGI ประเภทหนึ่งก็ได้ ดังแสดงดังภาพที่ 2.34



ภาพที่2.34 หลักการทำงานของ ASP

2.5.4 ความต้องการของระบบในการใช้งาน ASP

2.5.4.1 เครื่องคอมพิวเตอร์ PC ความเร็ว CPU 486 ขึ้นไป ถ้ายิ่งสูงจะยิ่งดี (เริ่มต้นที่ Pentium 150MHz ขึ้นไป RAM 32 MB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4.2 ระบบปฏิบัติการ (Operating System) ที่เป็น Windows 95/98, Windows NT 4.0 (Server/Workstation) หรือ Windows 2000

2.5.4.3 โปรแกรมที่ทำหน้าที่จำลองเครื่องไมโครคอมพิวเตอร์เป็น เว็บเซิร์ฟเวอร์ ซึ่งได้แก่

- PWS (Personal Web Server) เป็นเว็บเซิร์ฟเวอร์ที่สนับสนุนการทำงานบนระบบปฏิบัติการ Windows 95/98 ซึ่งอาจจะมีประสิทธิภาพไม่สูงนักแต่ก็สะดวกในการพัฒนางานเพื่อใช้ในการทดสอบก่อนส่งไฟล์เอกสารขึ้นไปยังเว็บไซต์ และเนื่องจาก PWS สามารถทำงานได้บน Windows95/98 ทำให้สามารถจำลอง Windows ที่ใช้งานอยู่เป็นเซิร์ฟเวอร์ได้ทันที
- IIS (Internet Information Server) สำหรับระบบปฏิบัติการ Windows NT
- โปรแกรม ASP ที่สามารถใช้งานได้ทันทีที่ติดตั้ง PWS4 หรือ IIS4 ขึ้นไป
- โปรแกรมที่ทำหน้าที่เป็นเว็บบราวเซอร์ เช่น Internet Explorer เป็นต้น

2.5.5 รูปแบบการเชื่อมต่อฐานข้อมูล

ในการติดต่อกับแฟ้มข้อมูลที่เป็น Access (*.mdb) หรือ Foxpro(*.dbf) ASP จะอาศัย ADODB (ActiveX Data Object Database) เพื่อติดต่อและจัดการกับข้อมูลในฐานข้อมูลนั้น สำหรับขั้นตอนในการติดต่อแฟ้มข้อมูลของ ASP ที่ต้องการติดต่อกับแฟ้มข้อมูล Access จะต้องอาศัยไดร์เวอร์ของแฟ้มข้อมูลนั้นช่วยซึ่งมี 3 วิธีคือ

- ODBC (Open DataBase Connectivity)
- DSNLess
- OLEDB

โดยจะกล่าวถึง 2 วิธีคือ ODBC และ DSNLess โดยทั้ง 2 แบบใช้ ADODB

2.5.5.1 การติดต่อฐานข้อมูลโดยใช้ ODBC

การติดต่อฐานข้อมูลแบบนี้จะต้องมีการกำหนดชื่อ DSN (Data Source Name) ให้กับ ODBC โดยจะต้องกำหนดชื่อไฟล์ฐานข้อมูลและชนิดของไดร์เวอร์ว่าเป็นอะไร

ข้อดีสำหรับการติดต่อแบบนี้จะสะดวกถ้าต้องมีการเปลี่ยนฐานข้อมูลจาก Access เป็นอย่างอื่นเช่น SQL Server หรือ Oracle เป็นต้น เพราะสามารถใช้ชื่อเดิมแต่เปลี่ยนคุณสมบัติภายในเท่านั้น หลังจากนั้นก็สามารถใช้งานต่อไปได้ โดยไม่จำเป็นต้องแก้โปรแกรมใหม่ เพราะ ASP ไม่สนใจว่าภายในใช้ไดร์เวอร์อะไรอยู่ ดังนั้นคำสั่งเพื่อติดต่อฐานข้อมูล Access หรือ SQL Server จะเขียนเหมือนกัน คือ

```
Set ObjDB=Server.CreateObject("ADODB.Connection")
ObjDB.Open  DSN-Name , "" , ""
```

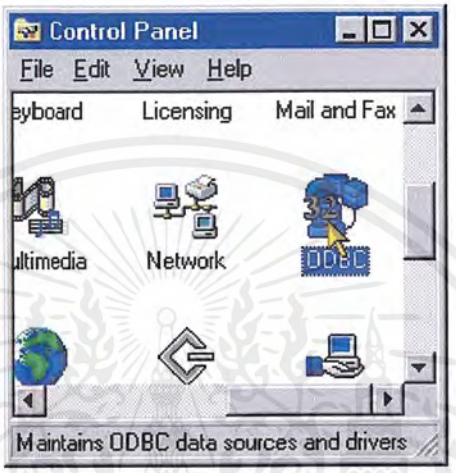
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ DSN-Name คือชื่อของ DSN ที่สร้างและกำหนดไดร์เวอร์เอาไว้แล้ว

ส่วนข้อเสียก็คือจะต้องไป Setup ODBC และสร้าง DSN ที่เซิร์ฟเวอร์เท่านั้น นั่นหมายความว่าจะต้องไปทำงานที่เซิร์ฟเวอร์นั้น

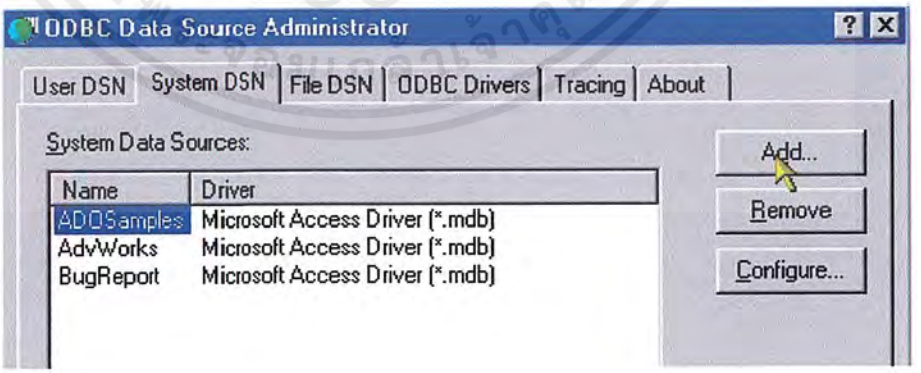
การสร้าง Setup ODBC และสร้าง DSN

1. เริ่มด้วยการเปิดโปรแกรม ODBC Data Source Administrator โดยการ Double-Click บน icon ของโปรแกรมที่อยู่ในหน้าต่าง Control Panel ดังภาพที่ 2.35



ภาพที่ 2.35 Icon ODBC บนหน้าจอ Control Panel

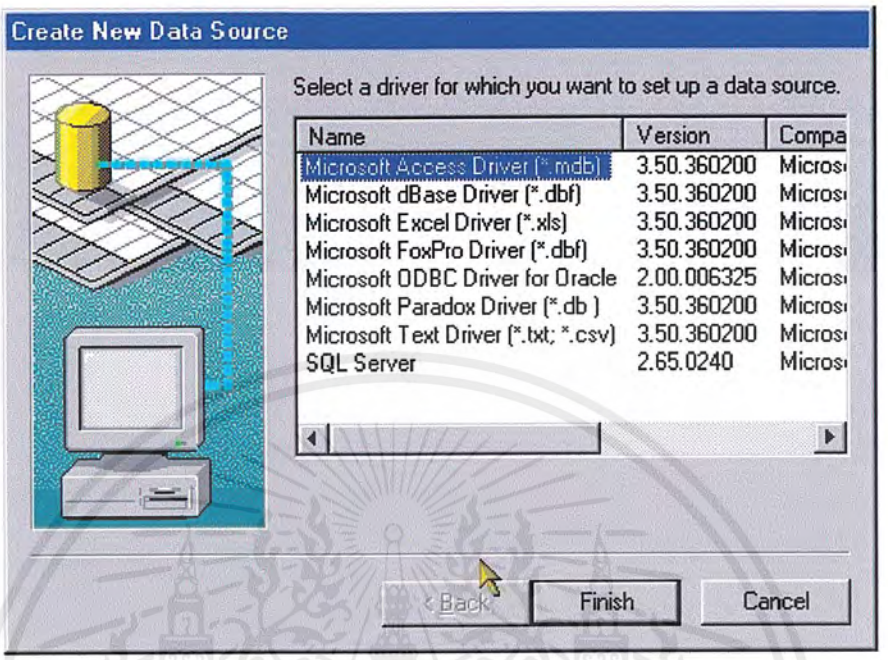
2. ใน ODBC Data Source Administrator ไดอะล็อค (dialog) ให้เลือก System DSN แท็ป(Tab) เพื่อที่เราจะได้ ตั้งค่า Data Source Name (DSN) ที่สามารถให้ผู้ใช้ทั้งหมดบน เครือข่ายเข้าใช้ฐานข้อมูลได้ ในการตั้งค่า(set up) ให้กับ System DSN ตัวใหม่ กดปุ่ม Add ดังภาพที่2.36



ภาพที่ 2.36 เลือก Tab Data Source Name

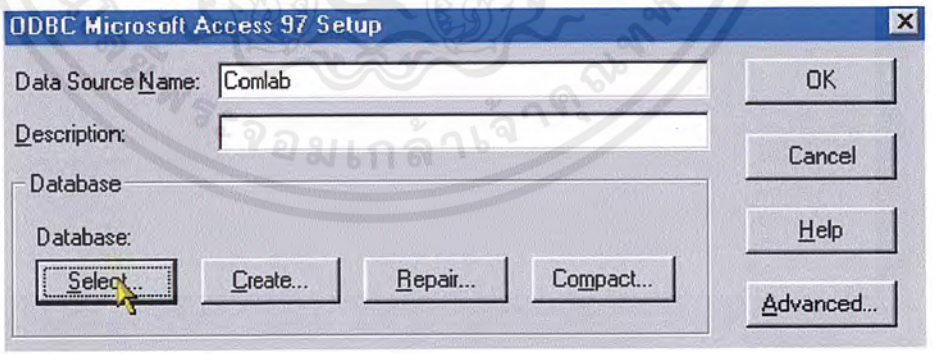
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นเลือก Driver ที่จะใช้กับ Data Source ในที่นี้เราต้องการติดต่อกับฐานข้อมูลของ Access ดังนั้นเราจึง เลือก Microsoft Access Driver จากรายการ Driver ที่มี ดังภาพที่ 2.37



ภาพที่ 2.37 เลือก Microsoft Access Driver จากรายการ Driver

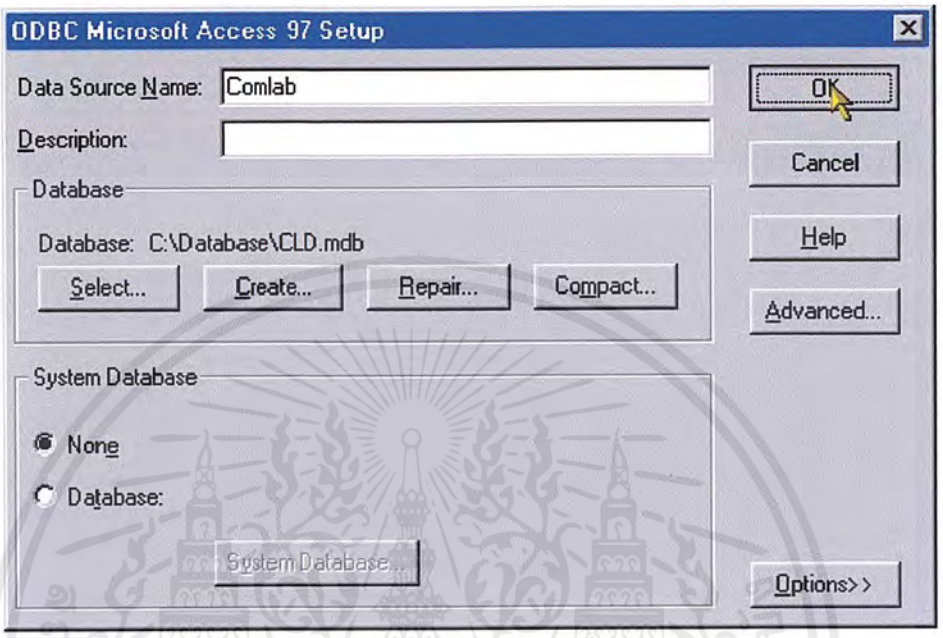
4. จากนั้นกดปุ่ม Finish จะเป็นการเปิด ODBC Microsoft Access 97 Setup ไดอะล็อค (dialog) ซึ่งจะเป็นที่ที่เราจะกำหนดชื่อของ Data Source (ชื่อนี้จะเป็นส่วนที่ Script ASP เรียกใช้ในการใช้งาน) และเราสามารถเลือกกำหนดพาส (Path) ของฐานข้อมูลได้โดย การกดปุ่ม Select ดังภาพที่ 2.38



ภาพที่ 2.38 หน้าจอ ODBC Microsoft Access 97 Setup ไดอะล็อค (dialog)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วเลือกฐานข้อมูลที่ต้องการใช้ จากนั้นกดปุ่ม OK เพื่อกลับเข้าสู่หน้าต่าง ODBC Data Source Administrator ไดอะล็อค มื่อเข้าสู่ ODBC Data Source Administrator ไดอะล็อค กดปุ่ม OK เพื่อสิ้นสุดการตั้งค่า System Data Source Name ดังภาพที่ 2.39



ภาพที่2.39 หน้าต่าง ODBC Data Source Administrator ไดอะล็อค

2.5.5.2 การติดต่อฐานข้อมูลโดยใช้ DSNLess

การติดต่อฐานข้อมูลแบบนี้จะไม่ใช้ DSN นั่นหมายความว่าไม่จำเป็นต้องสร้าง DSN หรืออาจกล่าวได้ว่าไม่มีการติดต่อกับ ODBC นั่นเอง

ข้อดีของการติดต่อแบบนี้ก็คือจะตัดขั้นตอนการเซ็ตอัพ ODBC ทั้งหมด ทำให้สะดวกในการใช้งานและไม่จำเป็นต้องเดินทางไปเซ็ตอัพที่เซิร์ฟเวอร์

ส่วนข้อเสียก็คือถ้าต้องการเปลี่ยนชนิดของฐานข้อมูล เช่นจาก Access เป็น SQL Server จะต้องไปแก้ไขรูปแบบการติดต่อทุกไฟล์ที่กล่าวถึงการติดต่อ

สำหรับรูปแบบการติดต่อแบบ DSNLess กับฐานข้อมูลที่ยกตัวอย่างคือ Access

```
Set ObjDB=Server.CreateObject("ADODB.Connection")
ObjDB.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ = Path and database"
```

โดยที่ path and database คือพาธและตำแหน่งของไฟล์ข้อมูลเนื่องจากใน 1 ฐานข้อมูล ประกอบด้วยออปเจ็กต์ เช่น ตาราง , คิวรีได้มากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการวิจัย

3.1 ขั้นตอนในการดำเนินงาน

3.1.1 ขั้นตอนการศึกษาได้แก่

3.1.1.1 ศึกษาการเขียนโปรแกรมบน Web

3.1.1.2 ศึกษาการพัฒนาโปรแกรมโดยใช้ Macromedia Flash 5 ในส่วนของการจำลองการจัดเรียง

3.1.1.3 ศึกษาเนื้อหาที่เกี่ยวข้องกับวิธีการจัดเรียงข้อมูลโดยอาศัยหลักการต่างๆ

3.1.1.4 ศึกษาวิธีการเขียนโปรแกรม ASP เพื่อใช้ในส่วนของแบบฝึกหัด

3.1.2 แบ่งองค์ประกอบต่างๆของโปรแกรมเป็นส่วนๆ

3.1.3 ทำการออกแบบและพัฒนาโปรแกรมในส่วนต่างๆ

3.1.4 นำโปรแกรมที่ได้ทำการพัฒนาในแต่ละส่วนมาประกอบกัน

3.1.5 ทดสอบการใช้งานโปรแกรมและทำการแก้ไขข้อบกพร่อง

3.1.6 จัดเตรียมคู่มือการใช้งานโปรแกม

3.2 คุณลักษณะของโปรแกรมที่จะออกแบบและพัฒนา

3.2.1 ผู้ใช้สามารถใช้งานโปรแกรมผ่านเครือข่ายอินเตอร์เน็ตได้เนื่องจากโปรแกรมถูกจัดทำขึ้นในรูปแบบของ Web Application

3.2.2 ผู้ใช้สามารถเลือกใช้งานโปรแกรมได้ 2 ภาษาคือ ภาษาอังกฤษ และ ภาษาไทย โดยภาษาหลักที่ ได้จัดเตรียมไว้คือภาษาอังกฤษ

3.2.3 เนื้อหาที่ได้จัดเตรียมไว้ถูกจัดเรียง และ แบ่งหัวข้อตามประเภทของหลักการที่ใช้ในการจัดเรียงข้อมูล อย่าง เป็นระเบียบ มีส่วนติดต่อผู้ใช้ที่สวยงามและสะดวกในการใช้งาน

3.2.4 โปรแกรมที่จัดทำขึ้นประกอบด้วยเนื้อหาของการจัดเรียงข้อมูลในแต่ละแบบพร้อมตัวอย่างสาธิตการจัดเรียงข้อมูลซึ่งได้จัดทำในรูปแบบของภาพเคลื่อนไหว (Animation) มีคำอธิบายขั้นตอนการทำงานอย่างเป็นลำดับด้วยภาษาที่ผู้ใช้สามารถทำความเข้าใจได้ง่าย และโปรแกรมจำลองการจัดเรียงข้อมูลที่สามารถรับค่าจากผู้ใช้โดยมีส่วนติดต่อกับผู้ใช้ที่สวยงามเพื่อให้ผู้ใช้สามารถใช้งานได้จริงโดยลักษณะเป็นภาพเคลื่อนไหว (Animation) แสดงให้เห็นผลลัพธ์ที่ได้จากการทำงานโดยวิธีการจัดเรียงวิธีต่างๆอย่างเป็นขั้นตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โปรแกรมที่จัดทำขึ้นประกอบด้วยส่วนของแบบฝึกหัดซึ่งจัดทำขึ้นอย่างเป็นระบบ เพื่อประโยชน์สูงสุดสำหรับผู้ใช้งาน โดยจัดทำทั้งส่วนของผู้ใช้และส่วนของผู้ดูแลระบบซึ่งสามารถทำการเพิ่มเติมและเปลี่ยนแปลงโจทย์และคำตอบได้สะดวกโดยผ่านทาง Application ที่ได้จัดเตรียมไว้ มีจุดประสงค์เพื่อประเมินความเข้าใจของผู้ใช้โปรแกรม

3.2.6 ง่ายและสะดวกต่อการใช้งาน

3.3 การกำหนดเนื้อหา

เนื้อหาทั้งหมดได้มีการจัดเรียงไว้ตามลำดับต่อไปนี้คือ

3.3.1 บทนำ(Introduction)

3.3.2 วิธีการคำนวณหาฟังก์ชัน Big – O

3.3.3 เนื้อหาวิธีการจัดเรียงลำดับข้อมูลทั้ง 10 ประเภทได้แก่ Selection Sort , Heap Sort , Bubble Sort ,Insertion Sort , Tree Sort , Quick Sort , Merge Sort , Shell Sort , Proxmap Sort , Radix Sort

โดยในแต่ละ Sort ประกอบด้วย

- รายละเอียดขั้นตอนการจัดเรียงลำดับข้อมูลของแต่ละวิธี
- Source Code ของวิธีการจัดเรียงลำดับข้อมูลแต่ละวิธีด้วยภาษา Pascal
- อธิบายตัวอย่างการใช้อัลกอริทึม
- ประสิทธิภาพของวิธีการจัดเรียงลำดับข้อมูลด้วยแต่ละวิธีนั้นๆ
- เปรียบเทียบประสิทธิภาพการทำงานของการจัดเรียงข้อมูลด้วยวิธีต่างๆ
- ตัวอย่างสาธิตการจัดเรียงข้อมูลด้วยโปรแกรม Animation พร้อมคำอธิบายการทำงานแต่ละขั้นตอนโดยละเอียด

3.3.4 แบบฝึกหัด

3.4 โปรแกรมประยุกต์ที่ใช้ในการพัฒนาโปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเตอร์เน็ต

3.4.1 Macromedia Flash 5.0

3.4.2 Macromedia Dreamweaver 4.0

2.4.3 โปรแกรม ASP(Active Server Page)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

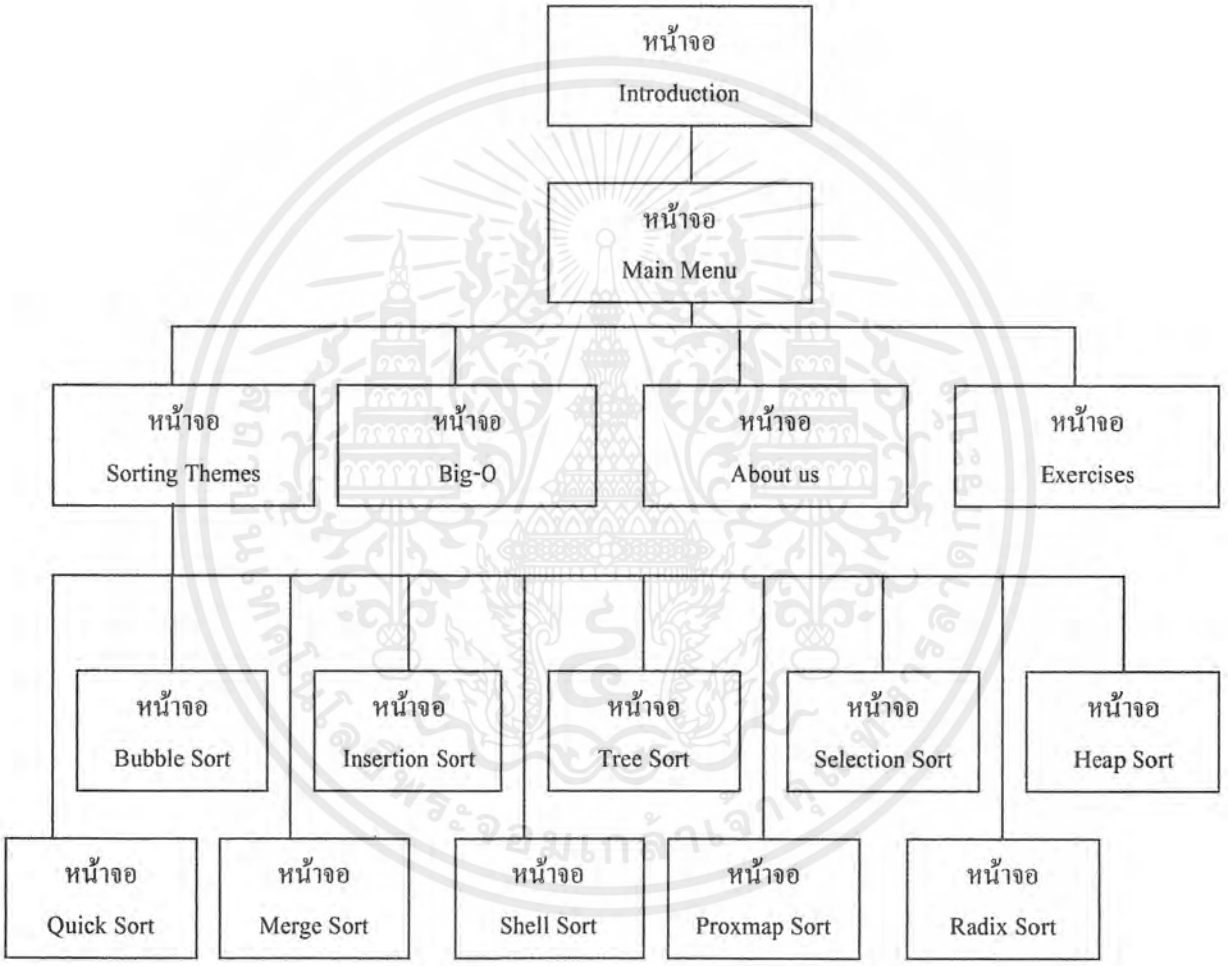
3.5 อุปกรณ์ที่ใช้ในการพัฒนาโปรแกรม

3.5.1 คอมพิวเตอร์ CPU Pentium III 733

3.5.2 คอมพิวเตอร์ หน่วยความจำ 128 MB

3.6 แผนภาพการทำงานของระบบ

3.6.1 โครงสร้างการทำงานของระบบโดยรวม



ภาพที่ 3.1 แสดงโครงร่างการทำงานของระบบโดยรวม

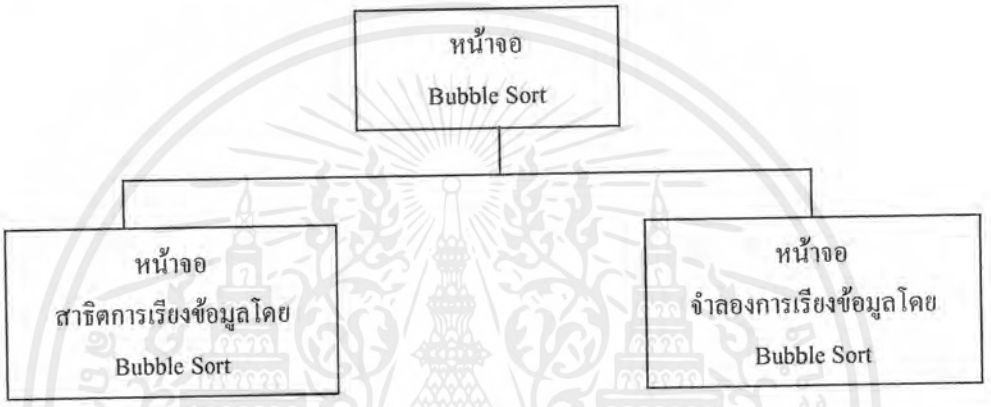
จากภาพที่ 3.1 หน้าจอแรกของโปรแกรมคือหน้าจอ Introduction ซึ่งได้จัดทำขึ้นอย่างสวยงามและน่าสนใจเพื่อดึงดูดและแนะนำผู้ใช้ให้เข้าสู่ระบบ โดยการกดปุ่มเพื่อเชื่อมโยงเข้าสู่หน้าจอ Main Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

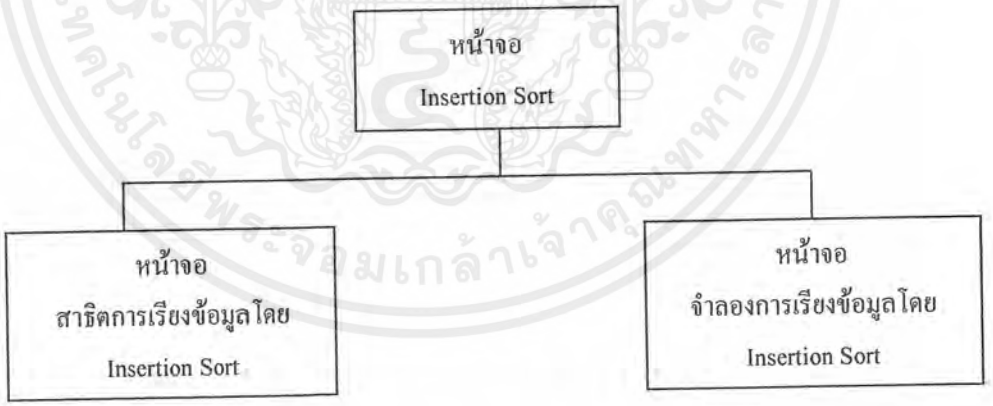
หน้าจอ Main Menu เป็นหน้าจอหลักของโปรแกรมซึ่งมีการจัดรูปแบบในลักษณะของ Frame โดยใน Frame ด้านซ้ายจะประกอบไปด้วย Link ต่างๆที่ผู้ใช้สามารถเลือกใช้งานได้ โดยแบ่งการใช้งานออกเป็น 4 ส่วนใหญ่ๆคือ

1. Link ไปยังส่วนของเนื้อหาการจัดเรียงข้อมูล ซึ่งประกอบด้วย 10 Sorting Theme
2. Link ไปยังส่วนของเนื้อหาการหาฟังก์ชัน Big-OH
3. Link ไปยังส่วนของระบบแบบฝึกหัด
4. Link ไปยังส่วนของการแนะนำคณะผู้จัดทำ

3.6.2 โครงสร้างการทำงานของหน้าจอเนื้อหาการจัดเรียงข้อมูลแต่ละประเภท



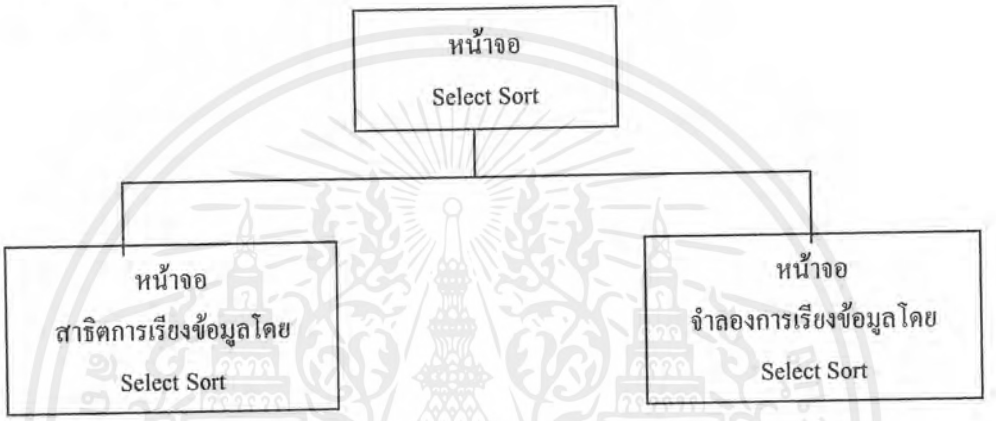
ภาพที่ 3.2 แสดงโครงร่างการทำงานของหน้าจอ Bubble Sort



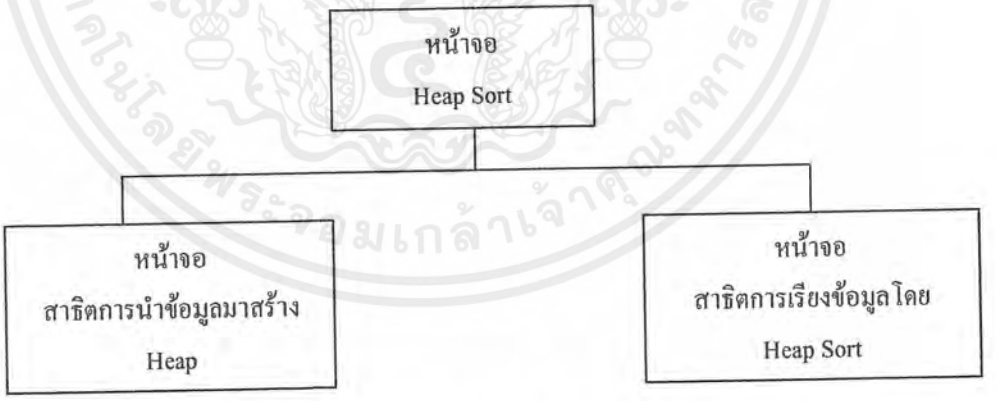
ภาพที่ 3.3 แสดงโครงร่างการทำงานของหน้าจอ Insertion Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.4 แสดงโครงร่างการทำงานของหน้าจอ Tree Sort

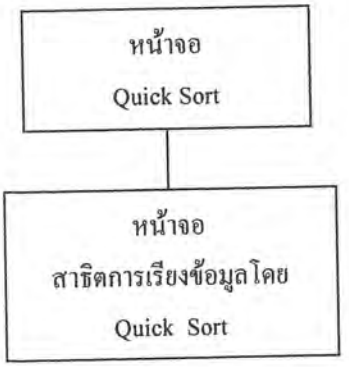


ภาพที่ 3.5 แสดงโครงร่างการทำงานของหน้าจอ Select Sort

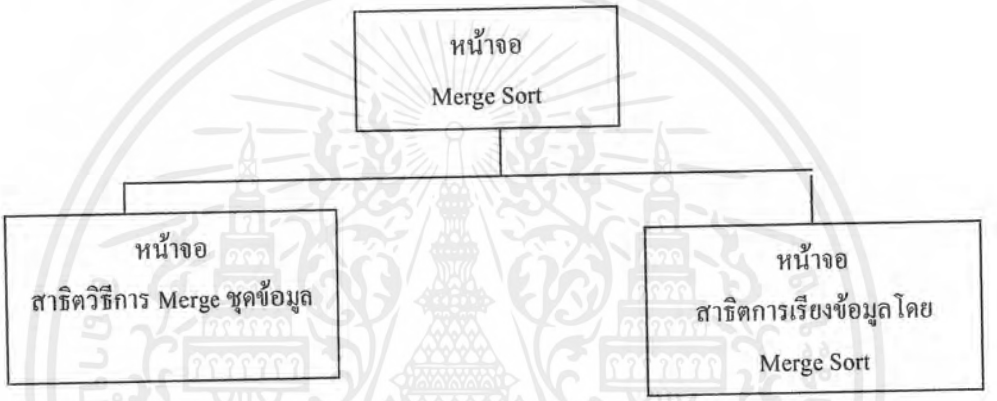


ภาพที่ 3.6 แสดงโครงร่างการทำงานของหน้าจอ Heap Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



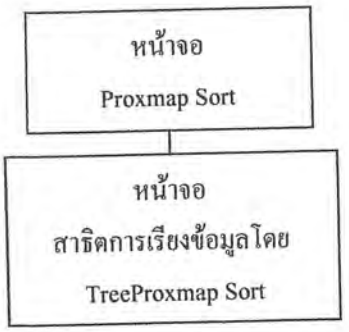
ภาพที่ 3.7 แสดงโครงร่างการทำงานของหน้าจอ Quick Sort



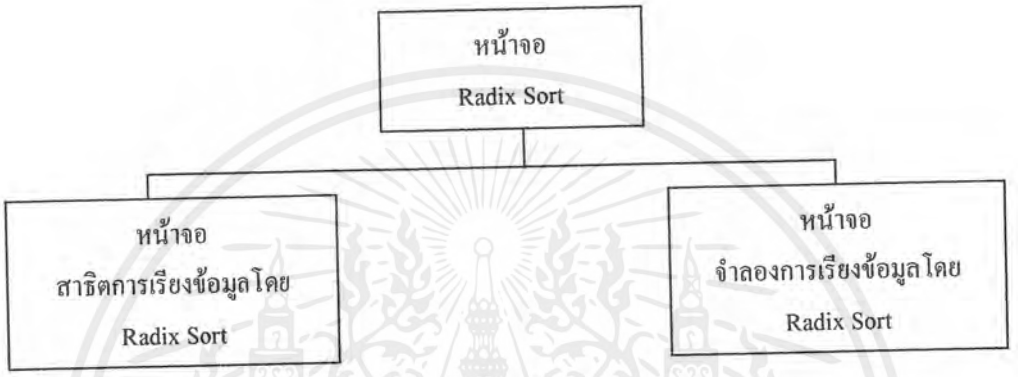
ภาพที่ 3.8 แสดงโครงร่างการทำงานของหน้าจอ Merge Sort

ภาพที่ 3.9 แสดงโครงร่างการทำงานของหน้าจอ Shell Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

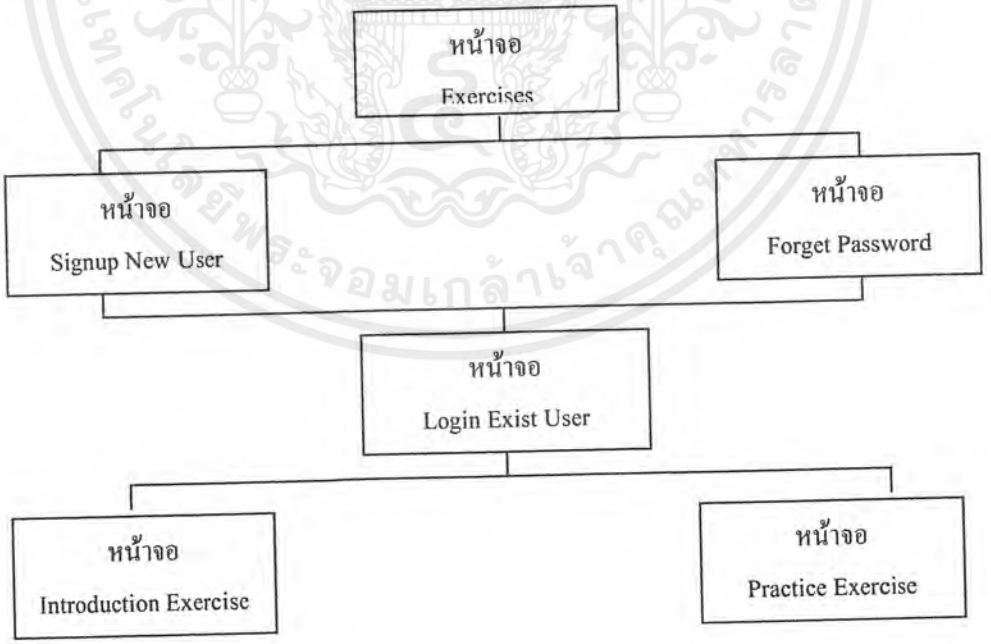


ภาพที่ 3.10 แสดงโครงร่างการทำงานของหน้าจอ Proxmap Sort



ภาพที่ 3.11 แสดงโครงร่างการทำงานของหน้าจอ Radix Sort

3.6.3 โครงสร้างการทำงานของหน้าจอระบบแบบฝึกหัด



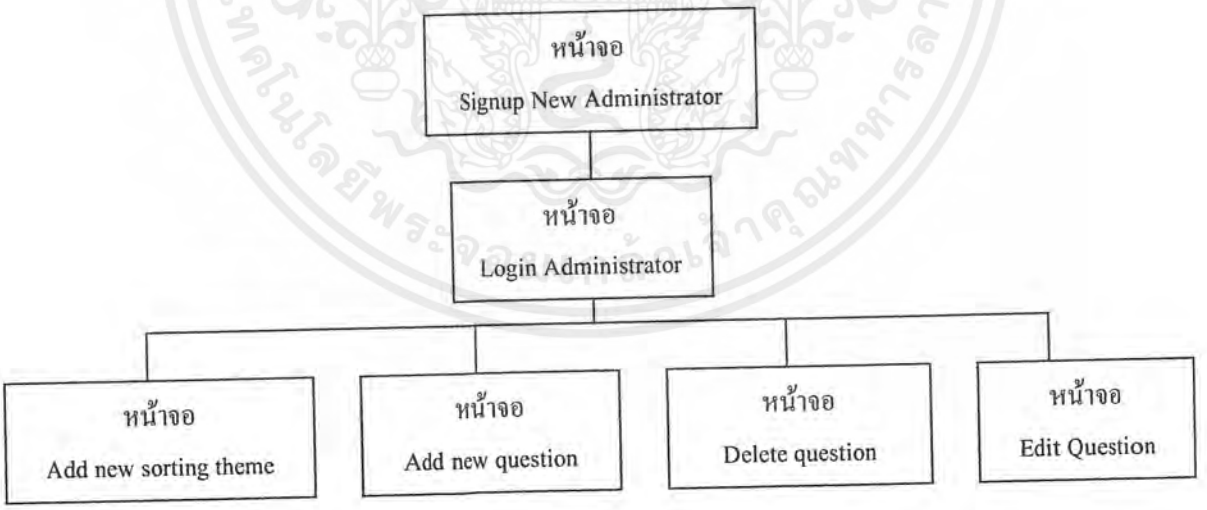
ภาพที่ 3.12 แสดงโครงร่างการทำงานของหน้าจอระบบแบบฝึกหัดในส่วนของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 3.12 เมื่อผู้ใช้เลือกใช้งานระบบแบบฝึกหัดจะปรากฏหน้าจอ Login User ซึ่งผู้ใช้จะต้องทำการใส่ User Name และ Password เพื่อได้รับอนุญาตในการเข้าใช้แบบฝึกหัดและหากผู้ใช้ยังไม่มี User Name และ Password ผู้ใช้จะต้องทำการ Signupเพื่อที่จะได้รับ User Name และ Password เพื่อผ่านเข้าสู่ระบบโดยผ่านหน้าจอ Sigup New User และในกรณีที่ผู้ใช้มี User Name และ Password แล้วแต่ลืม Password ของตนผู้ใช้สามารถที่จะเข้าสู่หน้าจอ Forget Password เพื่อระบบจะทำการแก้ปัญหาให้โดยการเปลี่ยน Password ซึ่งผู้ใช้จะต้องสามารถตอบคำถามจากคำถามที่ผู้ใช้ได้เคยตั้งไว้ในครั้งที่ทำการ ลงทะเบียนระบบครั้งแรก

เมื่อผู้ใช้มี User Name และ Password และทำการ Login เข้าสู่ระบบเรียบร้อยแล้วผู้ใช้จะสามารถเลือกใช้งาน แบบฝึกหัดซึ่งแบ่งเป็น 2 ประเภทคือ Introduction Exercise ซึ่งเป็นแบบฝึกหัดสำหรับทบทวนเนื้อหาความรู้พื้นฐานเกี่ยวกับการจัดเรียงข้อมูลซึ่งยังไม่เจาะจงวิธีการจัดเรียงด้วย Sorting Theme ใดๆ และ Practice Exercise ซึ่งเป็นแบบฝึกหัดที่ผู้ใช้จะสามารถทำการทดสอบความรู้ที่ได้จากการศึกษาเนื้อหาการจัดเรียงข้อมูลที่จัดเตรียมไว้โดยแยกตามประเภทของ Sorting Theme เมื่อผู้ใช้ได้ทำการเลือกใช้งานแบบฝึกหัดใดๆแล้วก็จะปรากฏหน้าจอของแบบฝึกหัดนั้นๆคือ หน้าจอ Practice Exercise และหน้าจอ Introduction Exercise

หน้าจอ Practice Exercise นั้นจะประกอบด้วยส่วนของ โจทย์และ หน้าต่างกระดาษคำตอบซึ่งเมื่อผู้ใช้ทำการ ตอบคำถามครบทุกข้อแล้วจะปรากฏหน้าจอแสดงคะแนนและประวัติการทำแบบฝึกหัดของผู้ใช้คนนั้น



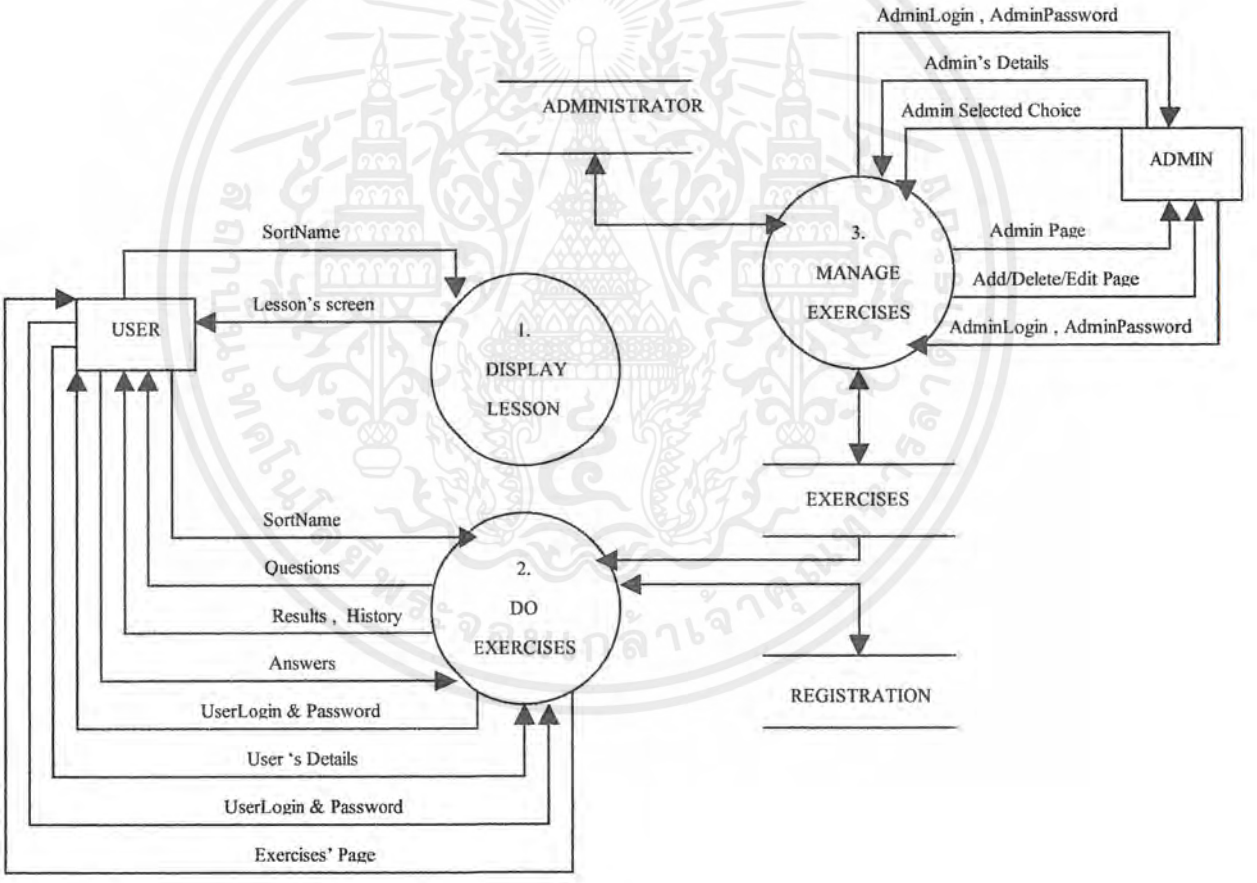
ภาพที่ 3.13 แสดงโครงร่างการทำงานของหน้าจอระบบแบบฝึกหัดในส่วนของ Administration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของหน้าจอสำหรับผู้ดูแลระบบแบบฝึกหัดนั้นจะไม่สามารถเข้าถึงได้จากหน้าจอหลักในส่วนของผู้ใช้เพื่อความปลอดภัย เมื่อผู้ดูแลระบบทำการ Login เข้าสู่ระบบผ่านหน้าจอ Login Administrator แล้วผู้ดูแลระบบสามารถใช้งานต่างๆของระบบโดยผ่านหน้าจอต่างๆได้แก่

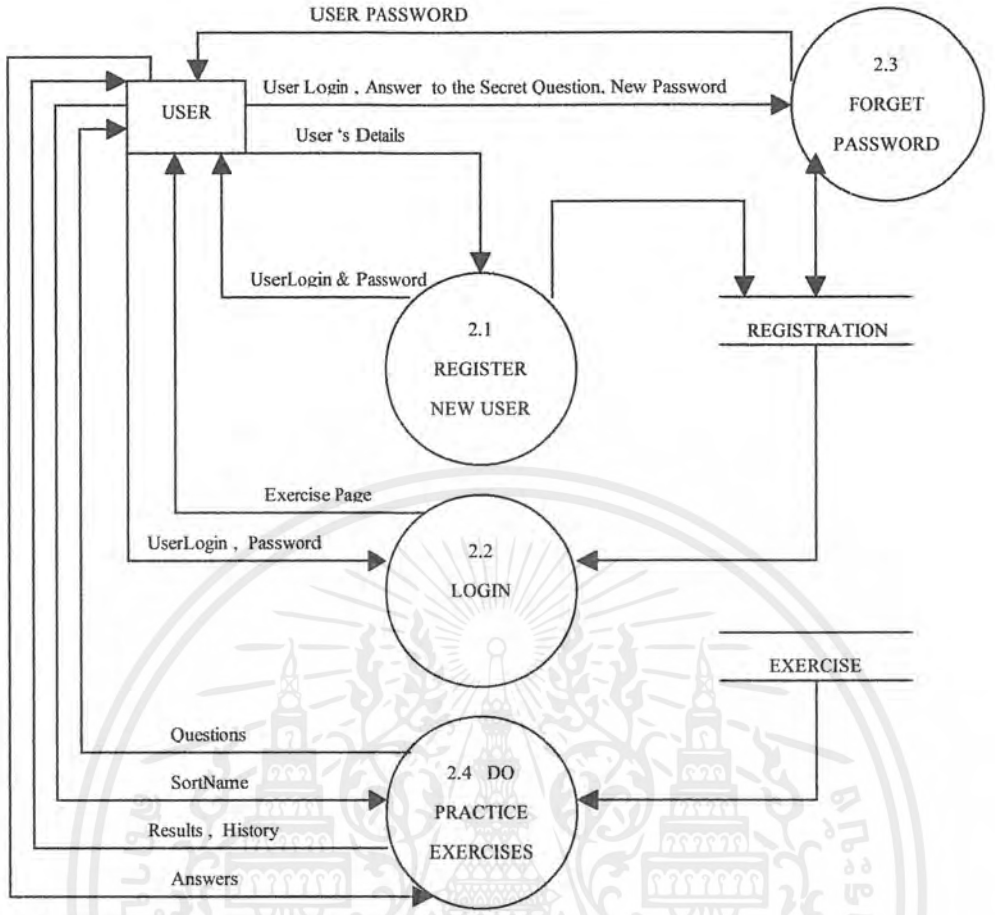
1. Add new sorting themeเพิ่มประเภทSorting Themeโดยผ่านทางหน้าจอAdd new Sorting Theme
2. Add new question เพิ่มจำนวนคำถามและทางเลือกของคำตอบในแต่ละSorting Theme
3. Delete question แก้ไขคำถามและทางเลือกของคำตอบในแต่ละ Sorting Theme
4. Edit question ลบคำถามและทางเลือกของคำตอบในแต่ละ Sorting Theme

3.6.4 แผนภาพแสดง Data Flow ของระบบ



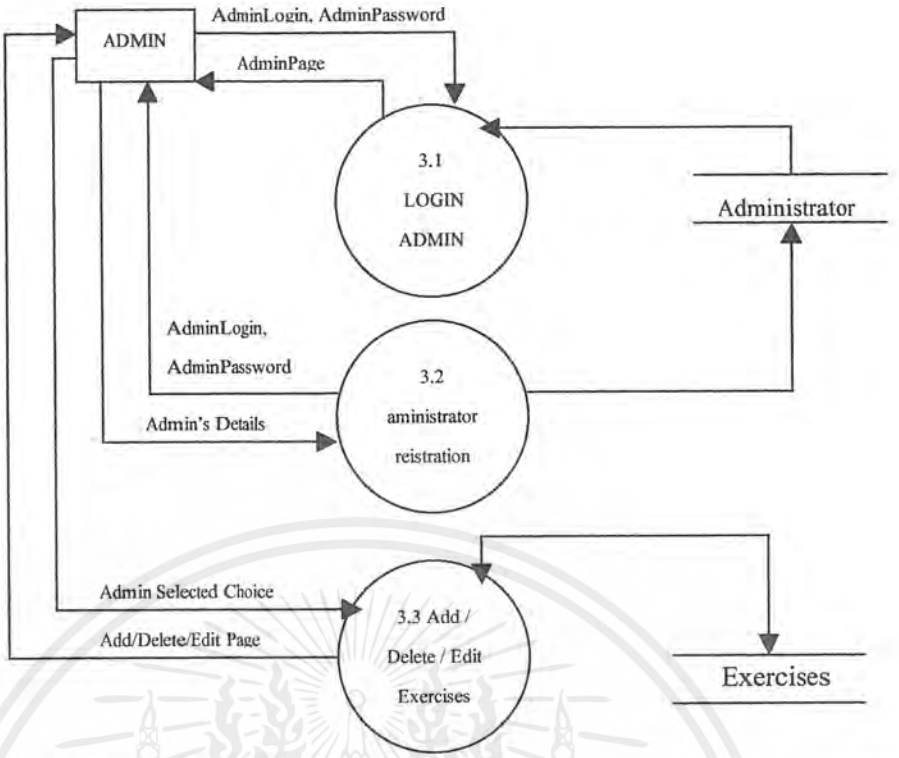
ภาพที่ 3.14 แสดง DFD ระดับ 0 ของระบบโปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.15 แสดง DFD ระดับ 1 ของ Process Do Exercises

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.16 แสดง DFD ระดับ 1 ของ Process Manage Exercises

3.6.5 Domain ของระบบงานแบบฝึกหัด

ระบบแบบฝึกหัดจัดทำขึ้นเพื่อให้ผู้ใช้โปรแกรมได้มีโอกาสทบทวนเนื้อหาและแนวความคิดในการจัดเรียงข้อมูลแต่ละวิธี ซึ่งได้มีการจัดทำขึ้นอย่างเป็นระบบโดยใช้ฐานข้อมูลในการ เก็บโจทย์ คำตอบ และเฉลย เพื่อความสะดวกในการเปลี่ยนแปลงและแก้ไขโจทย์ต่างๆและลดความซ้ำซ้อนในการเก็บข้อมูล อีกทั้งมีการบันทึกประวัติการเข้าใช้ระบบแบบฝึกของผู้ใช้แต่ละคน เพื่อเป็นข้อมูลให้ผู้ใช้ได้ทำการประเมินพัฒนาการความเข้าใจในเนื้อหา อีกทั้งเพื่อให้บรรลุจุดมุ่งหมายของโปรแกรมจำลองการจัดเรียงข้อมูล

ผู้ที่จะเข้าใช้บริการระบบแบบฝึกหัดต้องทำการลงทะเบียนเป็นสมาชิกซึ่งจะต้องให้รายละเอียดข้อมูล ต่างๆ ซึ่งได้แก่ ชื่อ (Name) , นามสกุล (Surname) , เพศ (Sex) , อายุ (Age) , ที่อยู่ (Address) , รหัสไปรษณีย์ (Zip Code) , อาชีพ(Ocupation) และ Email ซึ่งผู้ใช้จะเป็นผู้กำหนด User Login และ Password ด้วยตนเองและผู้ใช้จะต้องทำการตั้งคำถามและคำตอบที่เกี่ยวกับ Password นั้นเพื่อระบบจะใช้ในกรณีที่ผู้ใช้ลืม Password

ผู้ใช้บริการระบบแบบฝึกหัดสามารถใช้บริการต่างๆของระบบซึ่งได้แก่ การทำแบบฝึกหัดทั้งในส่วนของ Introduction Exercise และ Practice Exercise ซึ่งมีลักษณะแตกต่างกันกล่าวคือ Introduction Exercise จัดขึ้นเพื่อ เตรียมผู้ใช้บริการระบบแบบฝึกหัดให้มีความพร้อมเข้าสู่บทเรียน โดยจะรวบรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบฝึกหัดเกี่ยวกับบทเรียนโดยรวม ที่ไม่ต้องอาศัยพื้นฐานความรู้มาก โดยจะแบ่งออกเป็น Tsak ต่างๆ ซึ่ง เพิ่มจำนวนได้ตามผู้ควบคุมระบบซึ่งสามารถ เพิ่มชื่อ Task ได้โดยระบุ TaskNo และ TaskDetail แต่สำหรับในส่วนรายละเอียดภายในแต่ละ Task นั้นผู้ควบคุมระบบ จะต้องทำการ Implement ในส่วนของ Source Code เองเนื่องจากมิได้มีการเก็บเป็นฐานข้อมูลไว้ สำหรับส่วน Practice Exercise จัดทำขึ้นโดยแบ่งตามประเภทของ Sorting Theme ซึ่งผู้ควบคุมระบบสามารถเพิ่ม จำนวนแบบฝึกหัดดังกล่าวได้ โดยเริ่มด้วยการเพิ่มจำนวน Sorting Theme จากนั้นรายละเอียดของแบบฝึกหัดในแต่ละ Sorting Theme ผู้ควบคุมระบบสามารถทำการเพิ่มเติมได้โดยผ่าน Application ที่ได้จัดเตรียมไว้ อีกทั้งในการลดหรือเปลี่ยนแปลงแก้ไข ฐานข้อมูลแบบฝึกหัดนั้นผู้ควบคุมระบบสามารถทำงานโดยผ่าน Application ได้เช่นกัน เมื่อผู้ใช้ระบบทำแบบฝึกหัด Practice Exercise เรียบร้อยแล้ว ระบบจะทำการตรวจคำตอบพร้อมคำนวณคะแนน และแสดงประวัติการใช้งานระบบของผู้ใช้นั้นๆ

ผู้ดูแลระบบหรือ Admin จะทำหน้าที่ควบคุมระบบแบบฝึกหัด โดยสามารถ เพิ่มแบบฝึกหัดได้โดยผ่านทาง Application ซึ่งผู้ดูแลระบบจะต้องมีประวัติอยู่ในฐานข้อมูล ซึ่งได้แก่ ชื่อ (Name), Email, User Login, User Password โดย Admin จะมีสิทธิ์ในการเพิ่มจำนวนแบบฝึกหัด เพิ่ม Task เพิ่ม Sort อีกทั้งยังสามารถแก้ไขรายละเอียด และจำนวนแบบฝึกหัดได้อีกด้วย

3.6.6 รายละเอียดเกี่ยวกับตารางที่ใช้ในการพัฒนาระบบ

3.6.6.1. ฐานข้อมูล (Sortdb)

ตารางที่ 3.1 ตารางแสดงรหัสประจำแต่ละ Sorting Theme (Sort)

ENTITY	TYPE	KEY	DESCRIPTION
SortId	Number	PK	รหัสประจำSortingTheme
SortName	Text(20)		ชื่อ Sorting Theme

ตารางที่ 3.2 ตารางรายการ Introduction Exercise (IntroQuestion)

ENTITY	TYPE	KEY	DESCRIPTION
TaskNo	Number	PK	รหัสประจำแต่ละ Task
TaskDetail	Text(10)		ชื่อของ Task

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 ตารางแบบฝึกหัด (Exercise)

Description : แสดงรายละเอียดในส่วนของ โจทย์ และ คำตอบที่ถูกต้องในแต่ละข้อ

ENTITY	TYPE	KEY	DESCRIPTION
SortId	Number	PK	รหัสประจำ SortingTheme
Num	Number	PK	โจทย์ข้อ
Sentence	Text(50)		รายละเอียดของโจทย์
Rans	Text(1)		คำตอบที่ถูกต้อง

ตารางที่ 3.4 ตารางรายการคำตอบในแต่ละตัวเลือกของโจทย์แต่ละข้อ(Answer)

ENTITY	TYPE	KEY	DESCRIPTION
Run	AutoNumber	PK	
SortId	Number	FK	รหัสประจำSortingTheme
Num	Number	FK	โจทย์ข้อ
AnsId	Text(1)		สัญลักษณ์ประจำตัวเลือก
Ansdetail	Text(50)		รายละเอียดคำตอบ

ตารางที่ 3.5 ตารางบันทึกการทำแบบฝึกหัดของผู้ใช้โปรแกรม (Do)

Description : แสดงรายการบันทึกการทำแบบฝึกหัดของผู้ใช้โปรแกรม

ENTITY	TYPE	KEY	DESCRIPTION
Fname	Text(50)	PK	ชื่อ
Lname	Text(50)	PK	นามสกุล
SortId	Number	PK	รหัสประจำSortingTheme
Score	Number		คะแนน
Date	Date/Time	PK	วันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 ตารางบันทึกคะแนนชั่วคราว (TempScore)

Description : ตารางใช้งานเพื่ออำนวยในการเขียนโปรแกรมใช้เก็บคะแนนชั่วคราว

ENTITY	TYPE	KEY	DESCRIPTION
Seq	AutoNumber	PK	ลำดับ
Score	Number(Double)		คะแนน
SortId	Number		รหัส

Query Table ของฐานข้อมูล Sortdb

1. **Q2** => SortId , SortName , Num , Sentence , Rans
2. **Q4** => SortId , SortName , Num , AnsId1 , AnsDetail1
   AnsId2 , AnsDetail2 , AnsId3 , AnsDeatil3 , AnsId4 , AnsDetail4
3. **Q5** => Seq , Score , SortId

3.6.6.2 ฐานข้อมูล(Registeration)

ตารางที่ 3.7 ตารางรายการประวัติสมาชิก (Member)

ENTITY	TYPE	KEY	DESCRIPTION
Name	Text(25)	PK	ชื่อ
Surname	Text(25	PK	นามสกุล
Sex	Text(1)		เพศ
Age	Integer		อายุ
Address	Text(70)		ที่อยู่
Zip/Code	Text(7)		รหัสไปรษณีย์
Occupation	Text(2)	FK	การศึกษา
User	Text(10)		User Login
Password	Text(10)		User Password
Email	Text(30)		ที่อยู่ Email

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 ตารางบันทึกประวัติผู้ควบคุมระบบ (Admin)

ENTITY	TYPE	KEY	DESCRIPTION
Name	Text(50)	PK	ชื่อ
Email	Text(30	PK	ที่อยู่ Email
User	Text(10)		Admin Login
Password	Text(10)		Admin Password

3.6.6.3 ฐานข้อมูล(Temp)

ตารางที่ 4.9 ตารางบันทึกสถานะผู้ที่เข้าใช้ระบบอยู่ในขณะนี้ (TLogin)

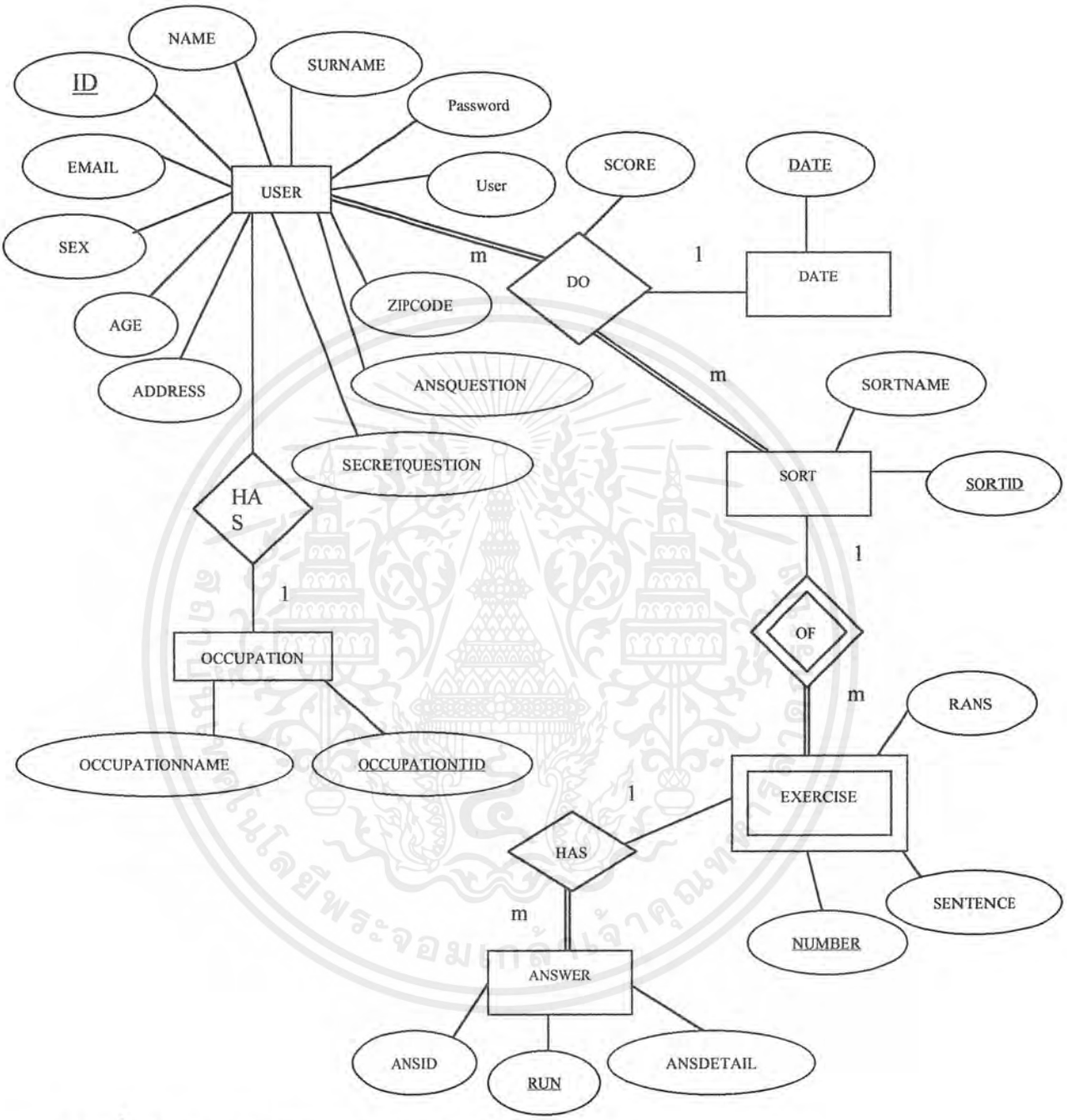
ENTITY	TYPE	KEY	DESCRIPTION
Name	Text(25)	PK	ชื่อ
Surname	Text(25)		นามสกุล

Query Table ของฐานข้อมูล Temp

1 **Q1** => Seq , Name , Surname , Email

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.7 แผนภาพแสดง E-R Diagram ของระบบแบบฝึกหัด



ภาพที่ 3.17 แสดง ER Diagram ของระบบแบบฝึกหัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

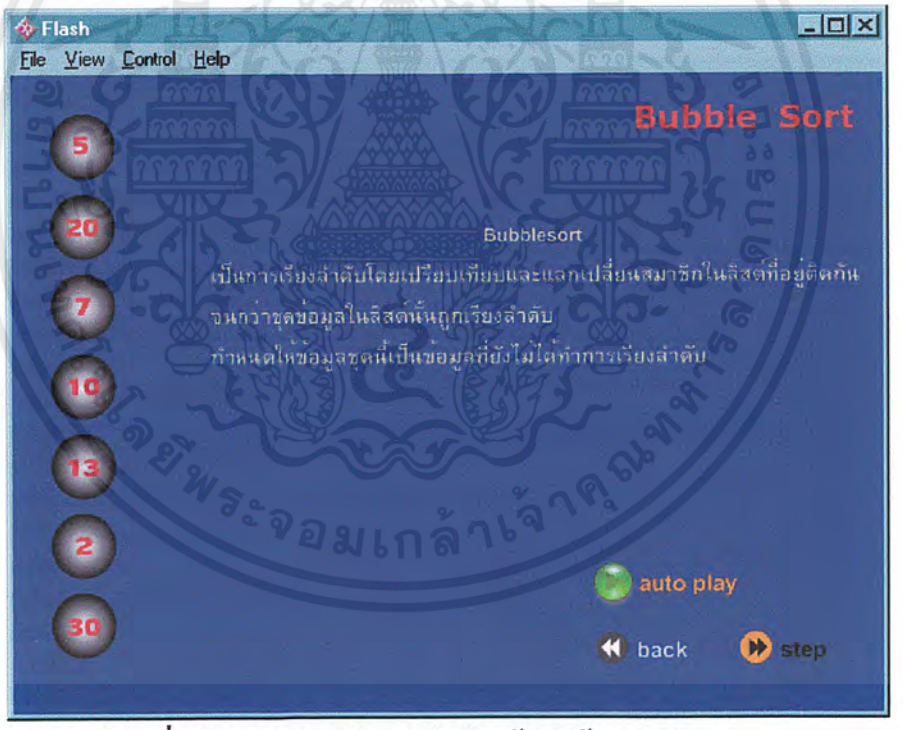
บทที่ 4
การพัฒนาระบบ

4.1 หลักการทำงานของโปรแกรมสาธิตการจัดเรียงข้อมูล

โปรแกรมสาธิตการจัดเรียงข้อมูลเป็นส่วนหนึ่งในการอธิบายเนื้อหาการจัดเรียงข้อมูลด้วยวิธีการต่างๆ โดยโปรแกรมสาธิตจะทำหน้าที่ช่วยเสริมความเข้าใจของการทำงานอย่างเป็นลำดับโดยอาศัยการยกตัวอย่างการจัดเรียงข้อมูลชุดหนึ่งๆ ที่โปรแกรมได้กำหนดขึ้น

4.1.1 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort

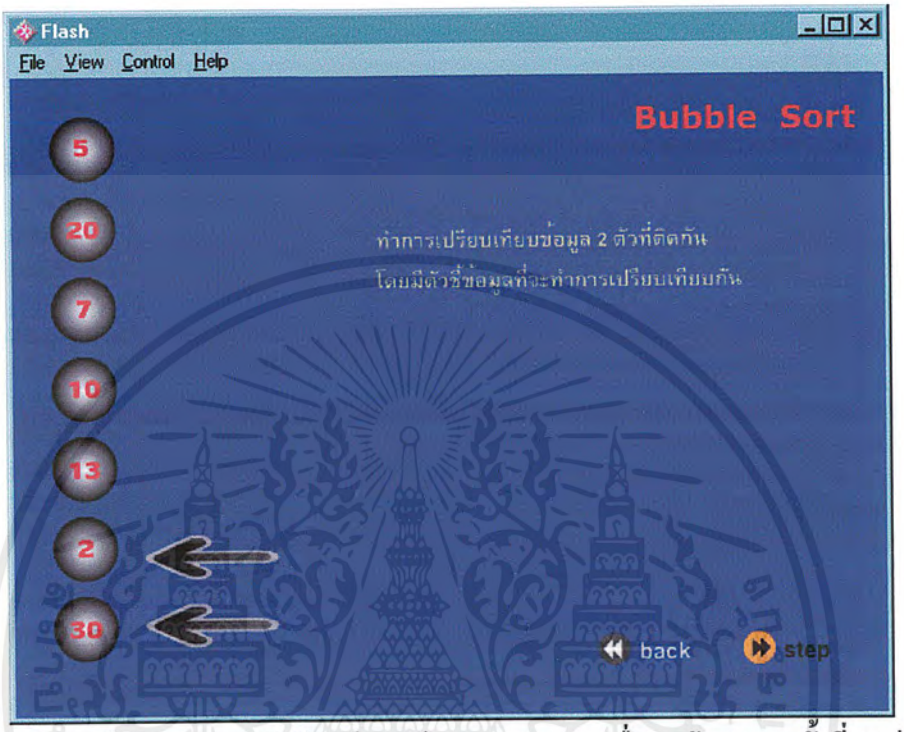
เมื่อพิจารณา Procedure Bubblesort จากภาพที่ 2.23 ซึ่งมีการทำงานในลักษณะของ Transposition Sorting โดยที่การจัดเรียงจะคล้ายกับการลอยตัวสูงขึ้นของฟองอากาศ ที่บางเบา นั่นคือค่าที่มีค่าน้อยที่สุดในแต่ละรอบจะถูกย้ายมาอยู่ที่ต้นของ ชุดข้อมูล



ภาพที่ 4.1 แสดงหน้าจอเริ่มต้นของการสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

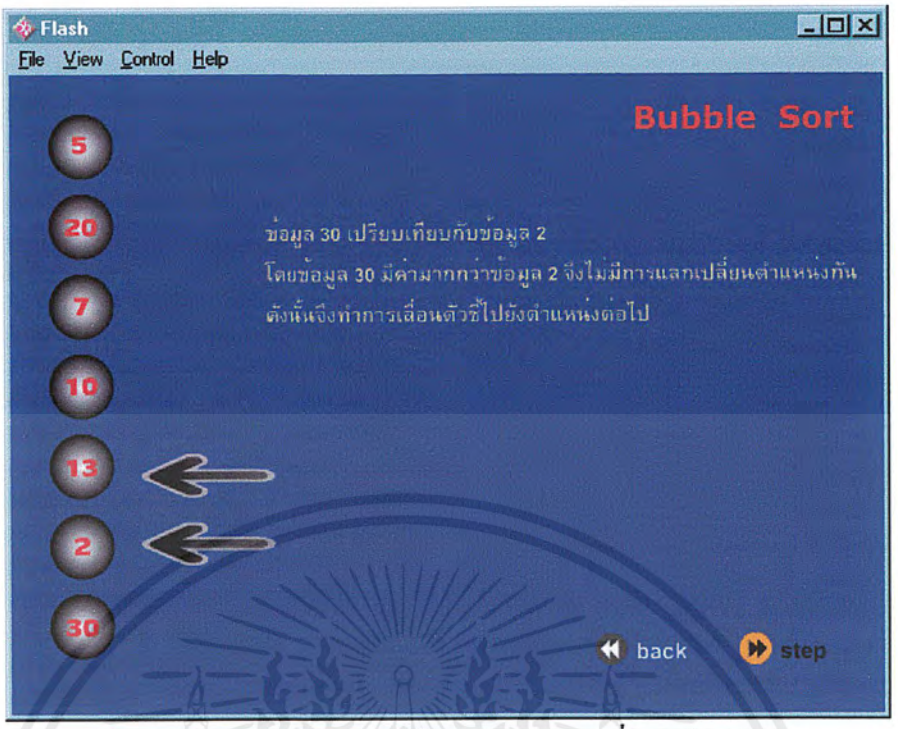
จากภาพที่ 4.1 เมื่อผู้ใช้กดปุ่ม Step ครั้งที่ 1จะแสดงขั้นตอนต่อไปของการทำงาน ในรอบที่ 1 (ค่า i=2 และ j=7)ของโปรแกรมนั่นคือจะแสดง Pointer ชี้ที่ตำแหน่ง a[6] และ a[7] และทำการเปรียบเทียบค่าทั้ง 2 ดังภาพที่ 4.2



ภาพที่ 4.2 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 1 (ค่า i=2 และ j=7) Pointer ชี้ที่ตำแหน่ง a[6] และ a[7] และทำการเปรียบเทียบค่าทั้งสองคือ 30> 2

กดปุ่ม Step ครั้งที่2 (ค่า i=2 และ j=6)Pointer ชี้ที่ตำแหน่ง a[5] และ a[6] และทำการเปรียบเทียบค่าทั้งสองคือ 2 < 13 ได้ผลลัพธ์ดังภาพที่ 4.3

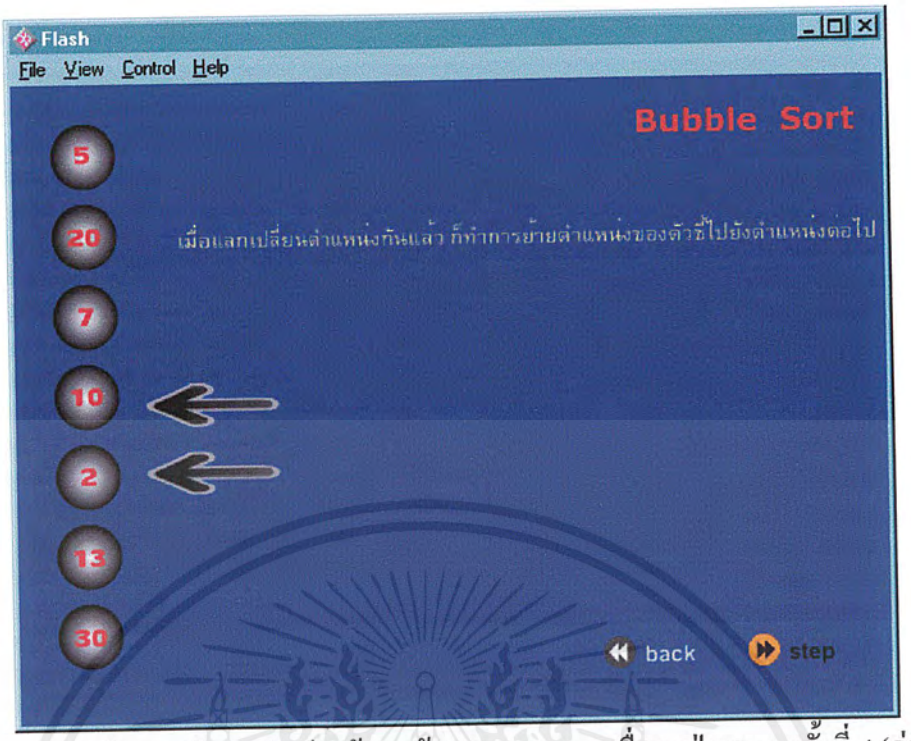
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



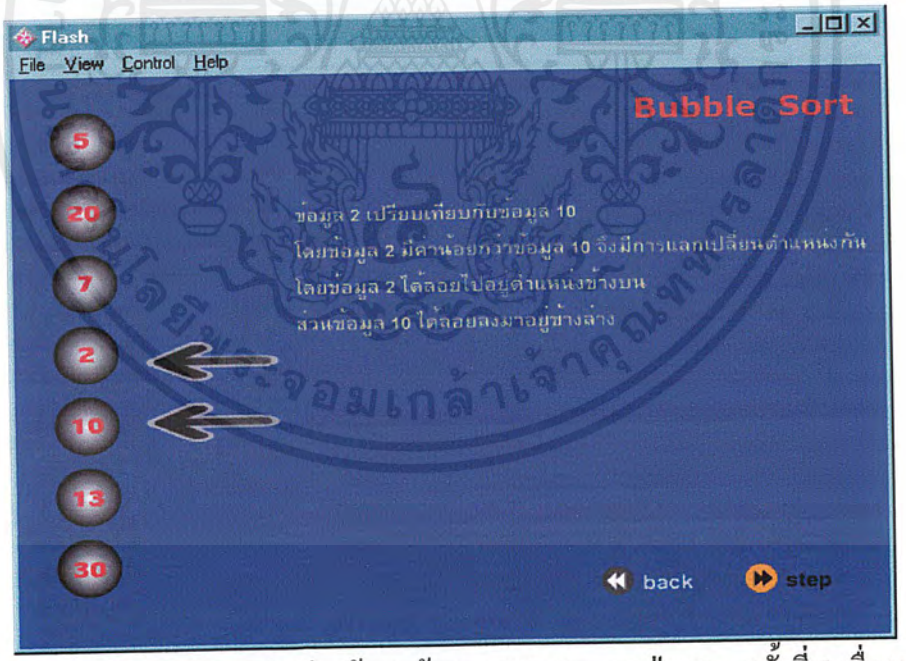
ภาพที่ 4.3 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่2 (ค่า i=2 และ j=6)

ภาพที่ 4.4 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 3 เนื่องจาก 2 น้อยกว่า 13 จึงมีการสลับค่าให้ค่า 2 ลอยสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

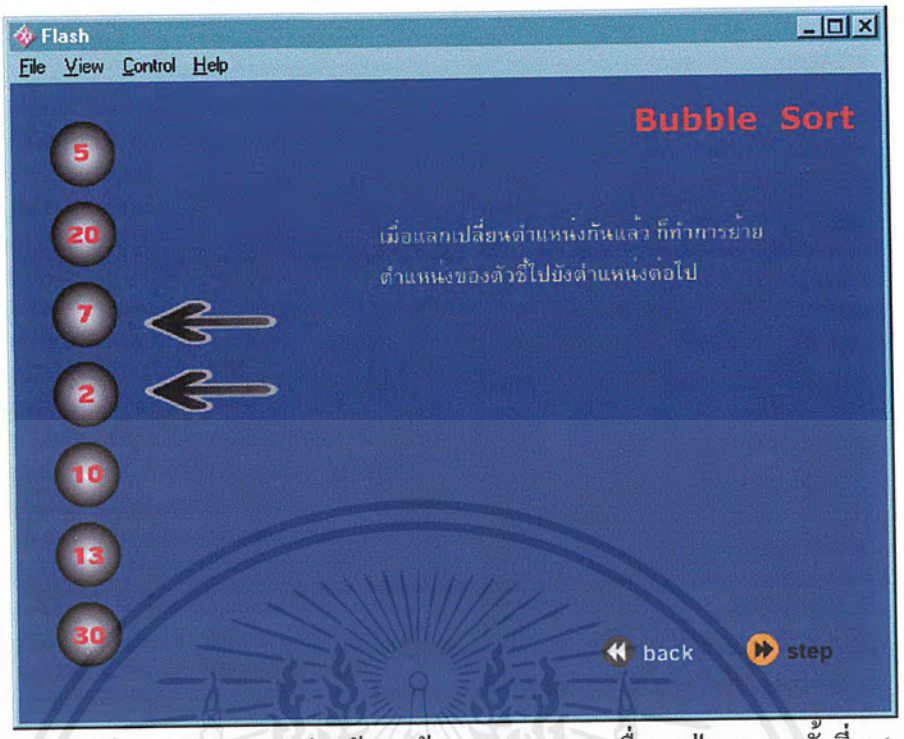


ภาพที่ 4.5 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 4 (ค่า i=2 และ j=5) และทำการเปรียบเทียบค่า ณ. ตำแหน่ง a[4] และ a[5] คือ 2 < 10

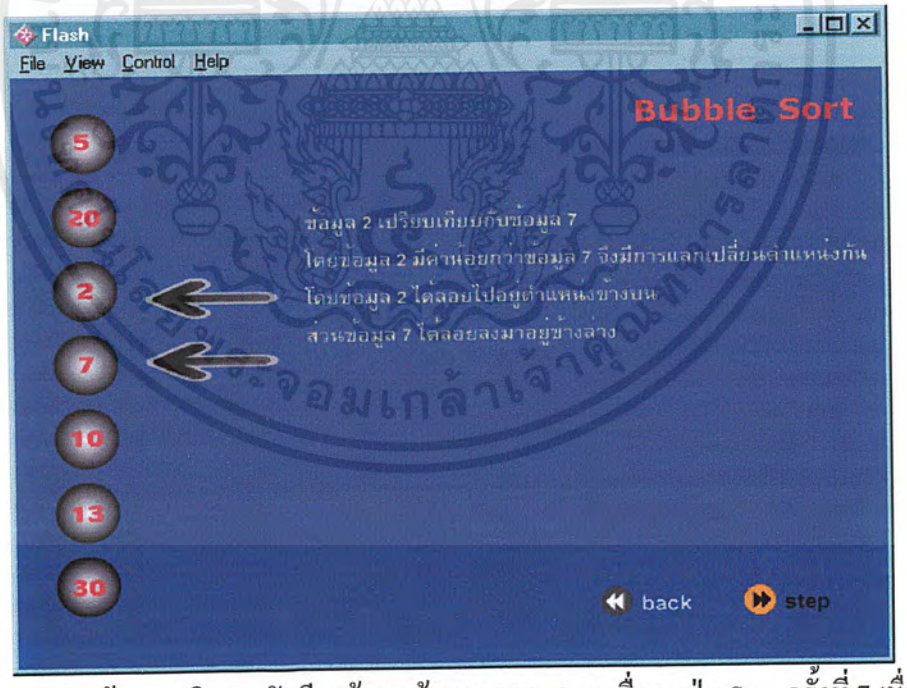


ภาพที่ 4.6 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort กดปุ่ม Step ครั้งที่ 5 เนื่องจาก 2 น้อยกว่า 10 จึงมีการสลับค่าให้ค่า 2 ลอยสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

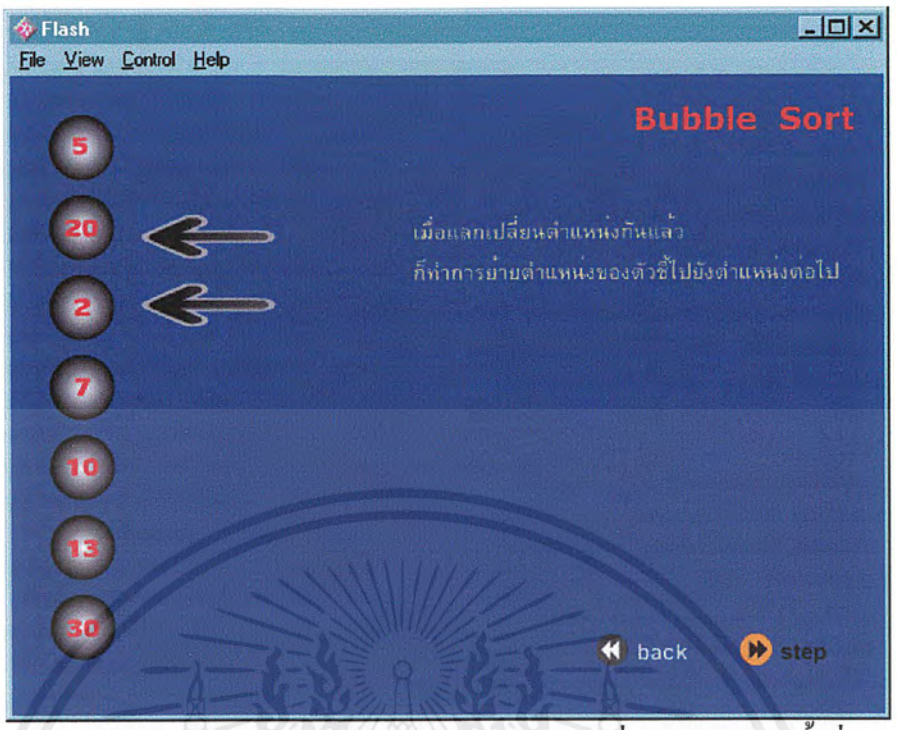


ภาพที่ 4.7 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 6 (ค่า I=2 และ j=4) และทำการเปรียบเทียบค่า ณ. ตำแหน่ง a[3] และ a[4] คือ 2 < 7

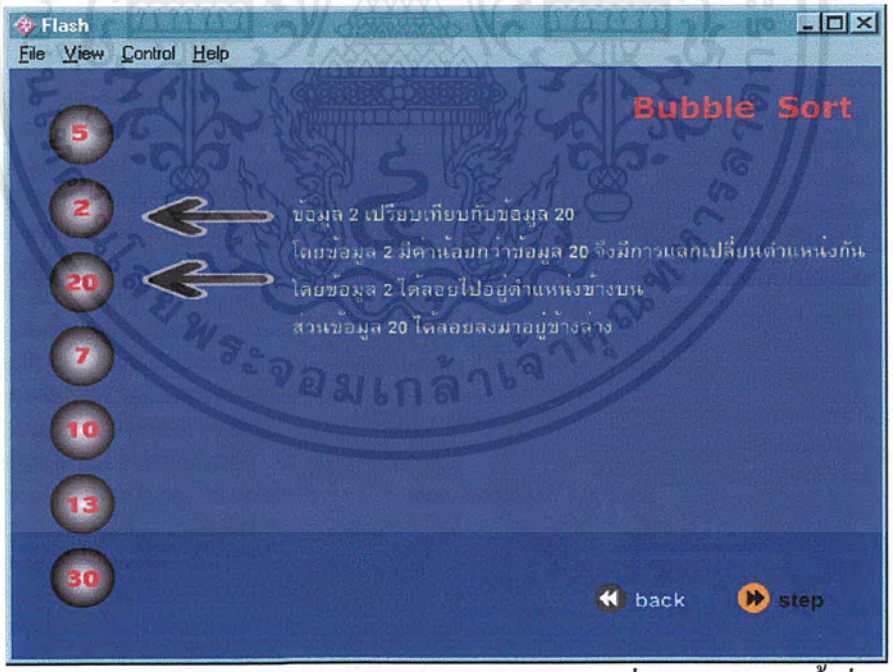


ภาพที่ 4.8 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 7 เนื่องจาก 2 น้อยกว่า 7 จึงมีการสลับค่าให้ค่า 2 ลอยสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

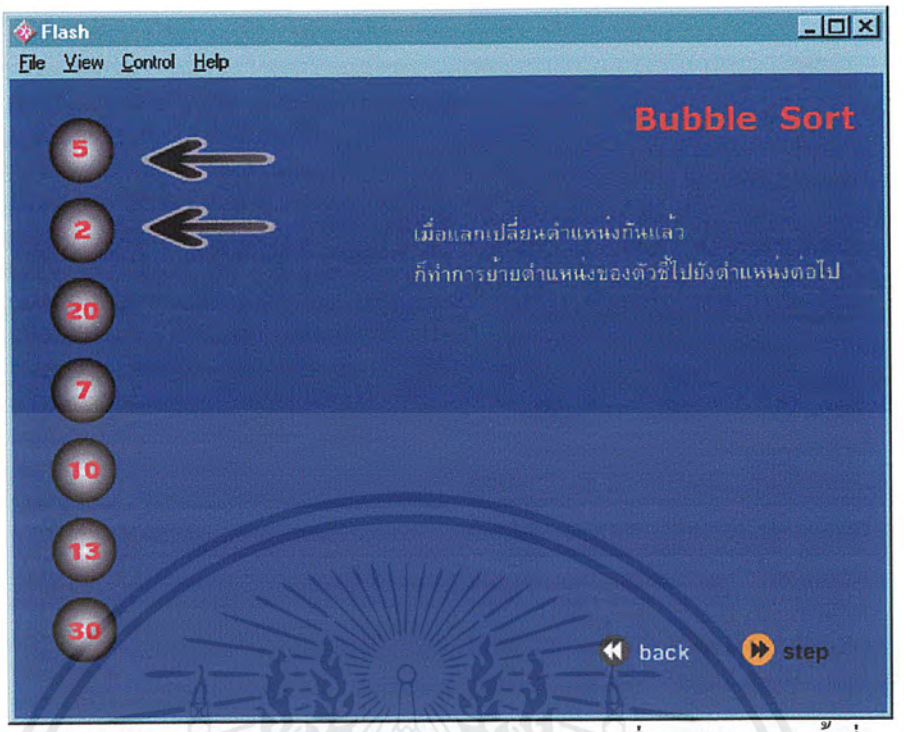


ภาพที่ 4.9 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 8 (ค่า I=2 และ j=3) และทำการเปรียบเทียบค่า ณ. ตำแหน่ง a[2] และ a[3] คือ 2 < 20

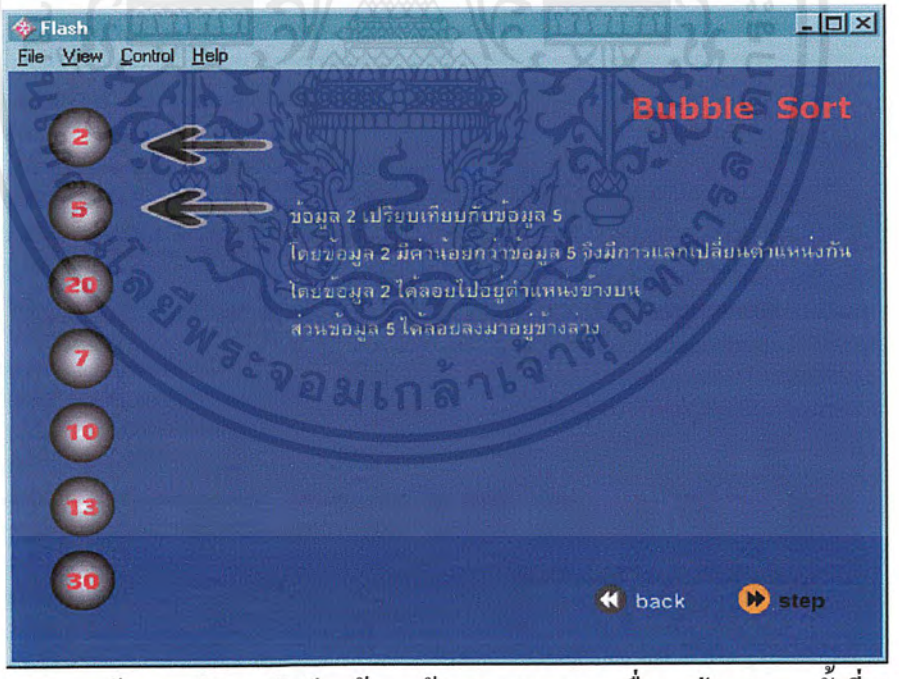


ภาพที่ 4.10 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 9 เนื่องจาก 2 น้อยกว่า 20 จึงมีการสลับค่าให้ค่า 2 ลอยสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

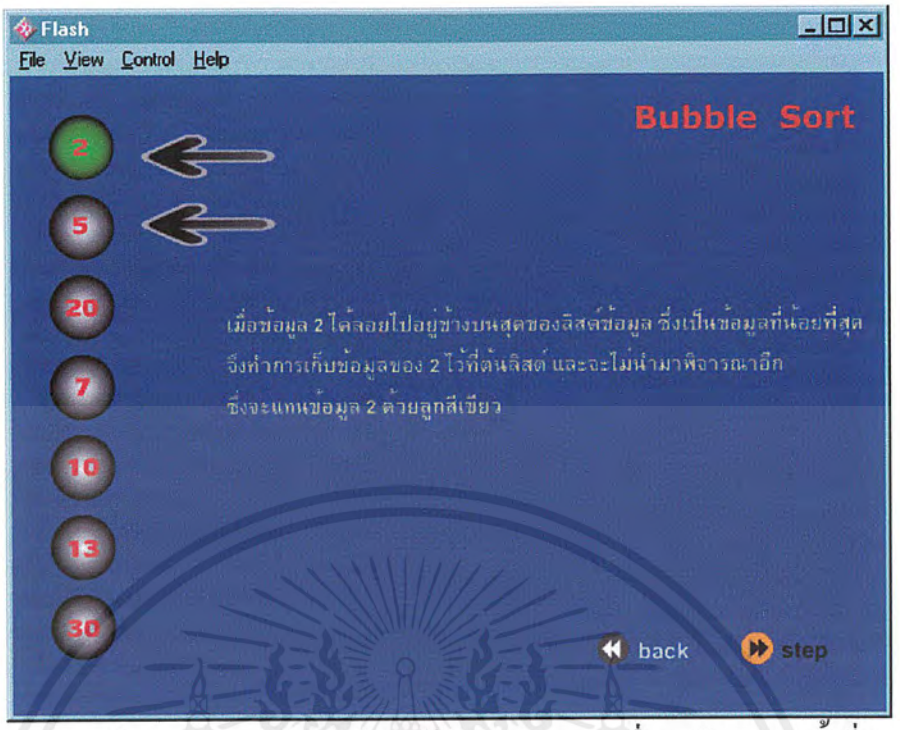


ภาพที่ 4.11 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 10 (ค่า I=2 และ j=2) และทำการเปรียบเทียบค่า ณ. ตำแหน่ง a[1] และ a[2] คือ 2 < 5



ภาพที่ 4.12 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 11 เนื่องจาก 2 น้อยกว่า 5 จึงมีการสลับค่าให้ค่า 2 ลอยสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

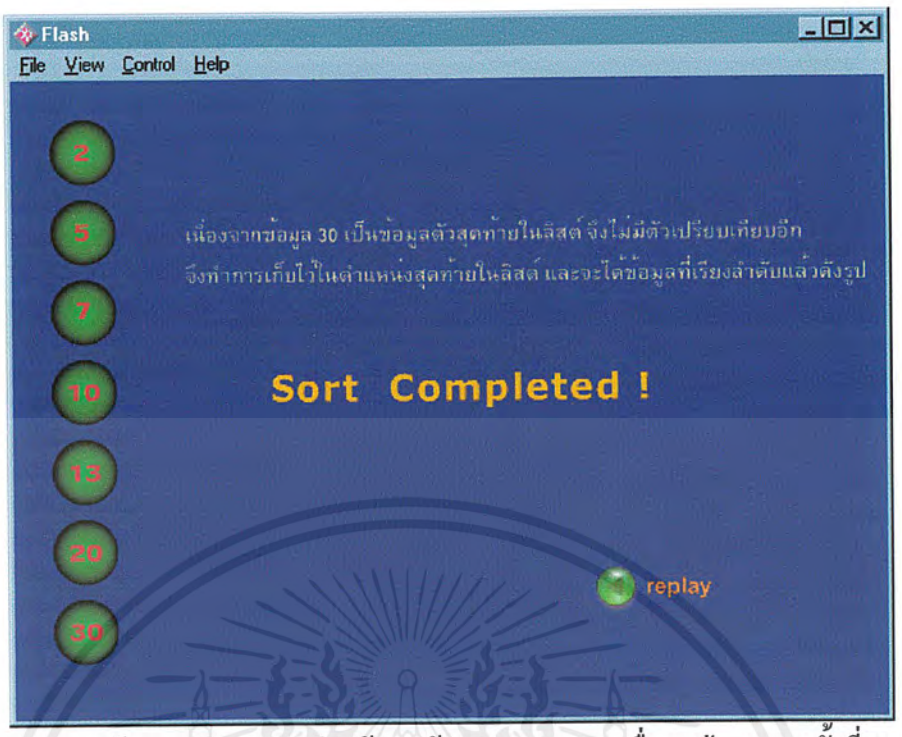


ภาพที่ 4.13 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 12 ตัวเลขค่าแรกจะเปลี่ยนเป็นสีเขียวนั่นคือสิ้นสุดการทำงานในรอบที่ i=2

เมื่อกดปุ่ม Step ในครั้งที่ 13 คือ i=3 และ j=7 ทำการเปรียบเทียบถ้า a[7] < a[6]

เมื่อกดปุ่ม Step ในครั้งที่ 14 ถ้า a[7] < a[6] ก็จะทำการสลับค่า a[7] และ a[6] มิเช่นนั้นแล้วก็จะทำการพิจารณาในรอบ i=3 และ j=6 ต่อไป ซึ่งจากชุดตัวอย่างนี้จะต้องทำการกดปุ่ม Step จำนวน 36 ครั้ง จะปรากฏหน้าจอดังภาพที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.14 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Bubble Sort เมื่อกดปุ่ม Step ครั้งที่ 36 จะแสดงหน้าจอสิ้นสุดการทำงานคือ Sort Complete

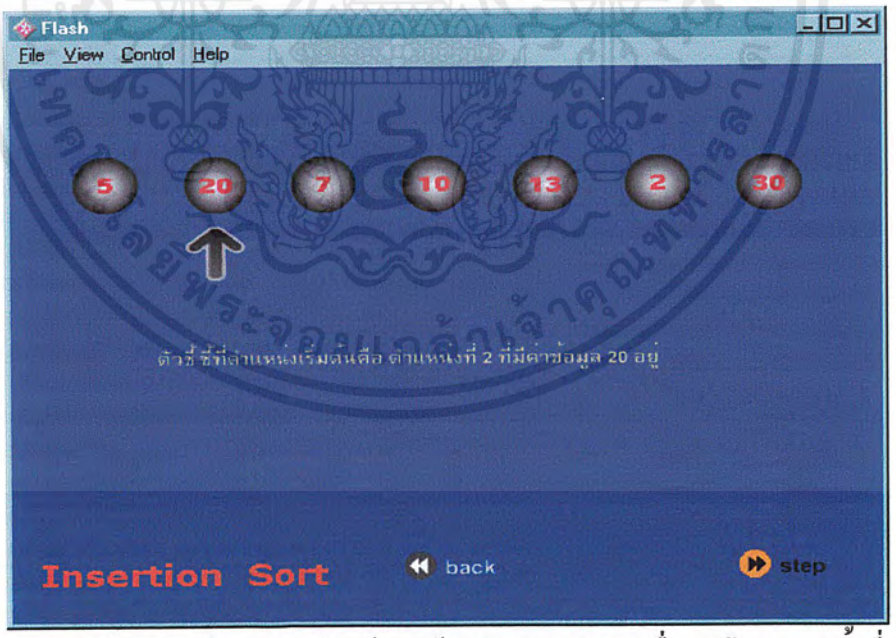
4.1.2 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort

เมื่อพิจารณา Procedure Insertion Sort จากภาพที่ 2.28 ซึ่งมีการทำงานในลักษณะของ Insert and keep Sorted ซึ่งมีลักษณะการทำงานที่ง่ายไม่ซับซ้อนเหมือนการจัดเรียงไพ่ตามเลขโดยการหยิบมาแทรก เช่น เลขชุดหนึ่งเป็น 5 20 7 10 13 2 30 มีจำนวนหมายเลขทั้งหมด 7 นั่นคือ n=7 ใช้array a[1..7] ในการเก็บข้อมูล และ ใช้a[0] เป็นเซ็นทิเนลซึ่งสามารถแสดงการจัดเรียงได้ดังภาพที่ 4.15 โดยชุดข้อมูลตัวอย่างคือ 5 , 20 , 7 , 10 , 13 , 2 , 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.15 แสดงหน้าจอเริ่มต้นของการสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort

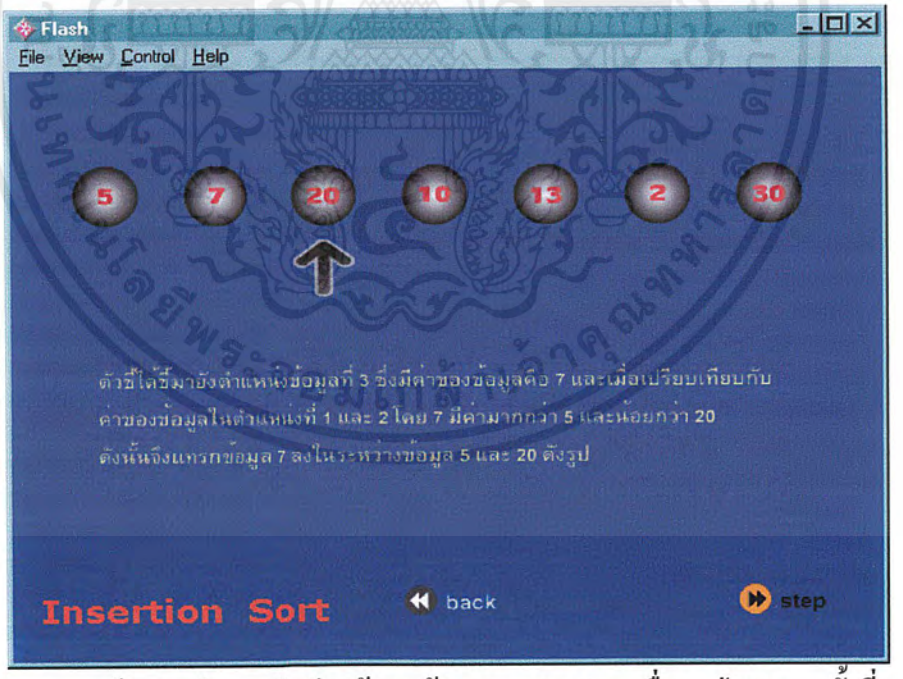
จากภาพที่ 4.15 เมื่อผู้ใช้กดปุ่ม Step ครั้งที่ 1จะแสดงขั้นตอนต่อไปของการทำงาน ในรอบที่ 1 (ค่า i=2 และ j=1) ของโปรแกรมนั่นคือจะแสดง Pointer ชี้ที่ตำแหน่ง a[2] ดังภาพที่ 4.16



ภาพที่ 4.16 แสดงหน้าจอ สาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 1จะมีการทำงานในรอบที่1(ค่า i=2 และ j=1) Pointer ชี้ที่ตำแหน่งa[2] และ a[0]=20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

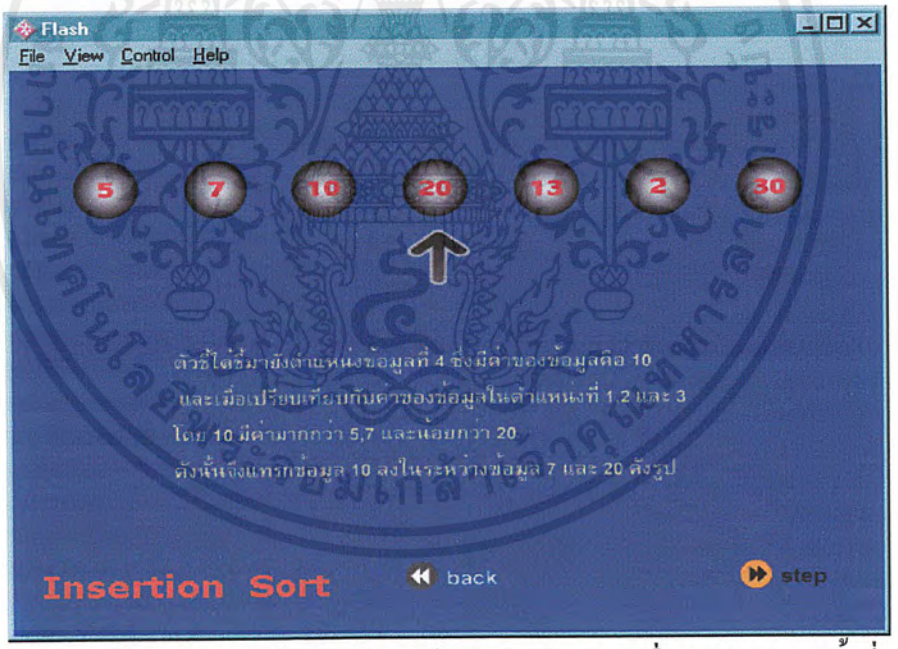
ภาพที่ 4.17 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 2 จะมีการทำงานในรอบที่ 2(ค่า I=3 และ j=2) Pointer ชี้ที่ตำแหน่ง a[3] และ a[0]=7



ภาพที่ 4.18 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 3 จะมีการแลกเปลี่ยนค่าระหว่างa[0] กับ a[2] และ a[0] กับ a[3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.19 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 4 จะมีการทำงานในรอบที่ 3(ค่า I=4 และ j=3)Pointer ชี้ที่ตำแหน่ง a[4] และ a[0]=10



ภาพที่ 4.20 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 5 จะมีการแลกเปลี่ยนค่าระหว่าง a[0] กับ a[3] และ a[0] กับ a[4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.21 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 6 จะมีการทำงานในรอบที่ 4(ค่า I=5 และ j=4) Pointer ชี้ที่ตำแหน่ง a[5] และ a[0]=13

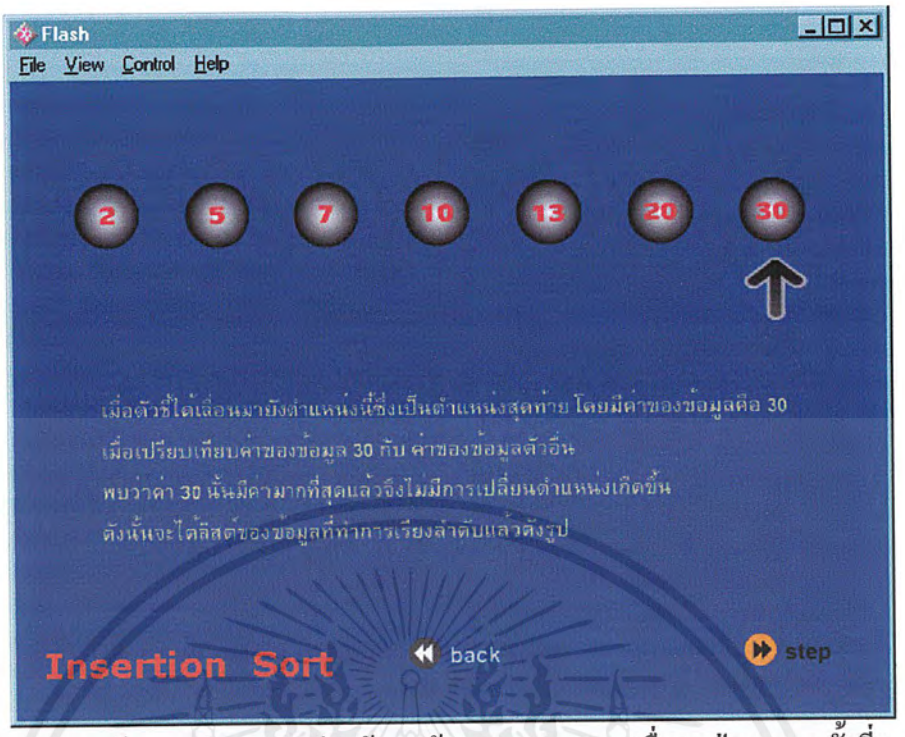
ภาพที่ 4.22 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่มครั้งที่ 7 จะมีการแลกเปลี่ยนค่าระหว่างa[0] กับ a[4]และ a[0] กับ a[5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

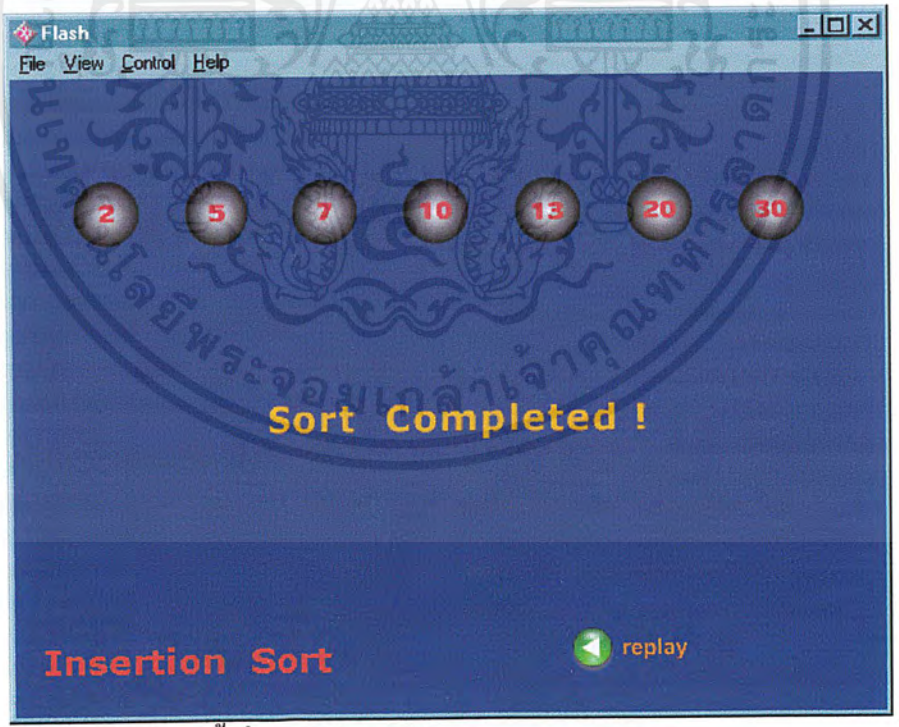
ภาพที่ 4.23 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 8 จะมีการทำงานในรอบที่ 5(ค่า I=6 และ j=5)Pointer ชี้ที่ ตำแหน่ง a[6] และ a[0]=2

ภาพที่ 4.24 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sort เมื่อกดปุ่ม Step ครั้งที่ 9 จะมีการแลกเปลี่ยนค่าระหว่างa[0] กับ a[5] และ a[0] กับ a[6] , a[0] กับ a[4] และ a[0] กับ a[5] , a[0] กับ a[3] และ a[0] กับ a[4] , a[0] กับ a[2] และ a[0] กับ a[3] , a[0] กับ a[1] และ a[0] กับ a[2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.25 แสดงหน้าจอสาธิตการจัดเรียงข้อมูลด้วย Insertion Sortเมื่อกดปุ่ม Step ครั้งที่ 10 จะมีการทำงานในรอบที่ 6(ค่า I=7 และ j=6) Pointer ชี้ ที่ตำแหน่ง a[7] และ a[0]=30



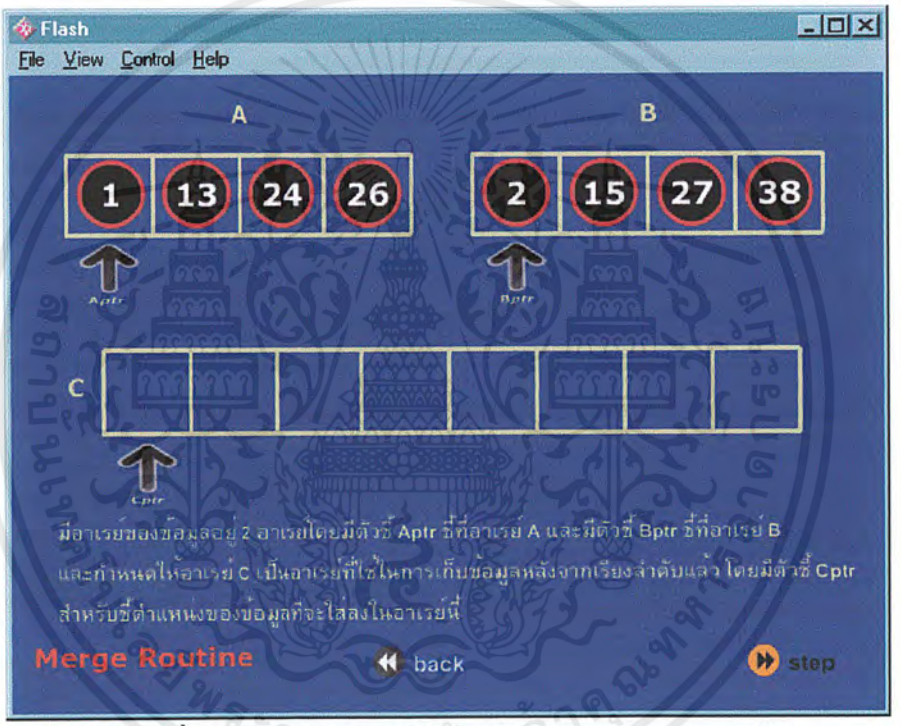
ภาพที่ 4.26 เมื่อกดปุ่ม Stepครั้งที่ 11 จะปรากฏข้อความ Sort Complete

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Merge Sort

เมื่อพิจารณา Procedure Merge Sort จากภาพที่ 2.21 ซึ่งมีการทำงานในลักษณะของ Divide-and-Conquer โดยมีแนวคิดในการแก้ปัญหาโดยแตกปัญหาออกเป็นกลุ่มๆและแต่ละกลุ่มมีจำนวนข้อมูลเท่ากันจนกระทั่งไม่สามารถแบ่งกลุ่มได้ แล้วทำการเรียงลำดับข้อมูลในแต่ละกลุ่มด้วยวิธีการ Merge Sort จากนั้นนำผลลัพธ์ที่เรียงลำดับแล้วมารวมกันจนได้ข้อมูลชุดเดิม

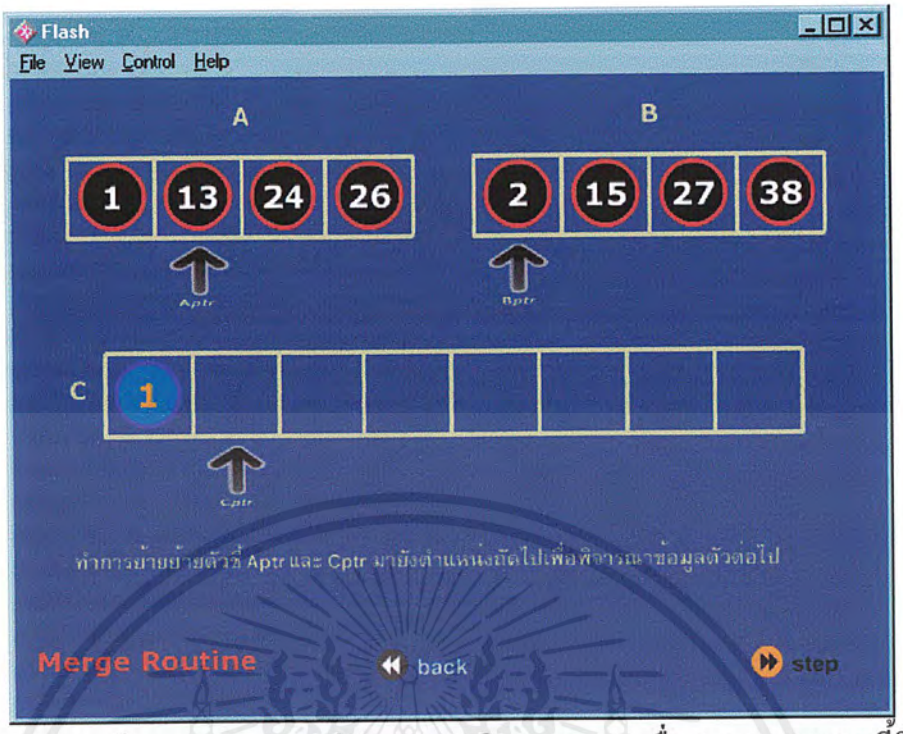
โดยโปรแกรมสาธิตการจัดเรียงข้อมูลด้วยวิธี Merge Sort นี้ในขั้นตอนแรกจะทำการสาธิตในส่วนของขั้นตอนการ Merge ข้อมูลในขั้นตอนสุดท้ายคือ Merge ชุดข้อมูล 1 13 24 26 และ 2 15 27 38 ให้เป็นชุดข้อมูล 1 2 13 15 24 26 27 38 ซึ่งสามารถแสดงได้ด้วยภาพที่ 4.27



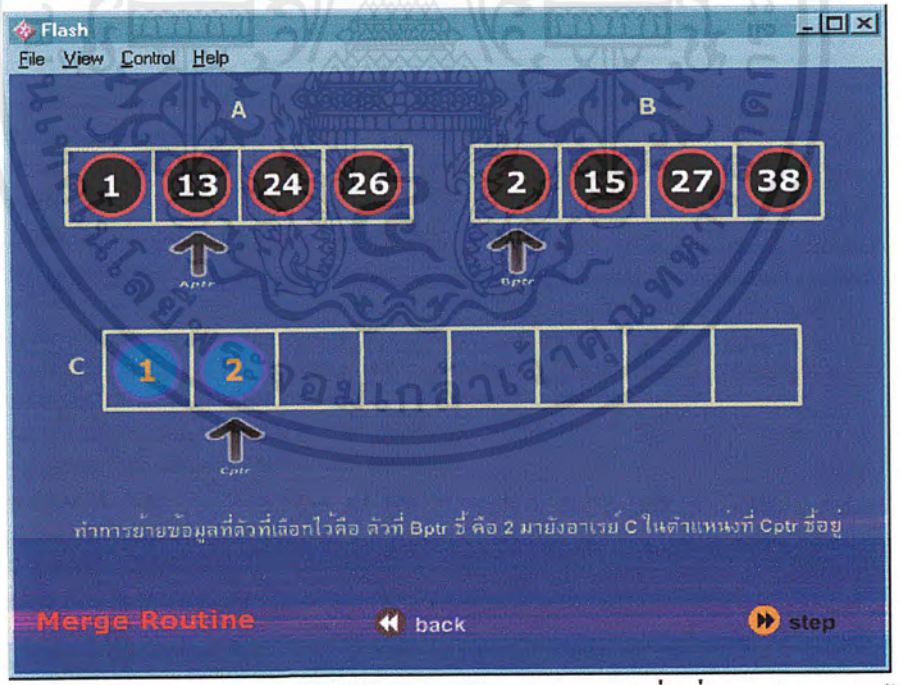
ภาพที่ 4.27 แสดงหน้าจอเริ่มต้นการทำงานของโปรแกรมสาธิตการ Merge ชุดข้อมูล

จากภาพที่ 4.27 เมื่อกดปุ่ม Step โปรแกรมจะทำการเปรียบเทียบค่าที่ index แรกของแต่ละกลุ่ม โดยจะเลือกค่าที่น้อยกว่ามาเก็บไว้ใน Array Cเนื่องจากเป็นการเรียงลำดับจากน้อยไปมากดังแสดงได้ดังภาพที่ 4.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



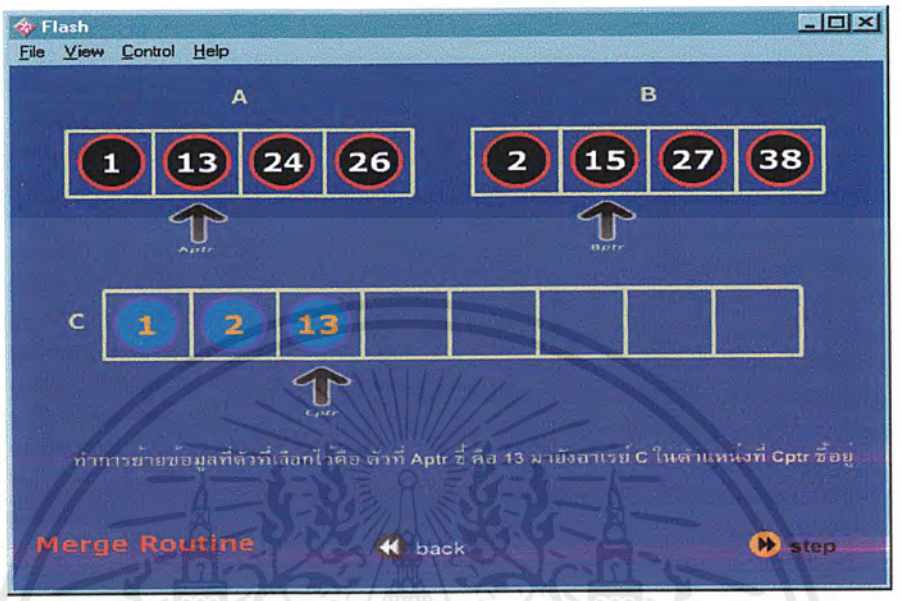
ภาพที่ 4.28 แสดงหน้าจอการทำงานของโปรแกรมสาธิตการ Merge เมื่อ Pointer Aและ Bชี้ที่ต้น List ของแต่ละกลุ่มทำการเปรียบเทียบเอาค่าที่น้อยกว่า



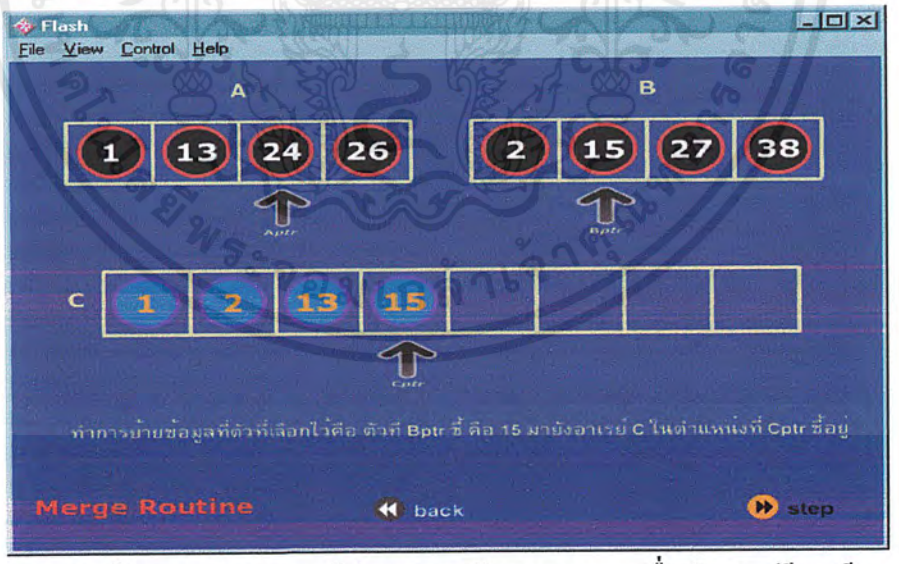
ภาพที่ 4.29 แสดงหน้าจอการทำงานของโปรแกรมสาธิตการ Merge เมื่อเลื่อน Pointer A,Cไปยัง Index ถัดไปทำการเปรียบเทียบค่าที่ Index Aและ B เลือกค่าน้อยมาเก็บไว้ใน C ที่ตำแหน่ง Index Cptr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 4.29 เลื่อน Pointer B,C ไปยัง Indexถัดไปแล้วทำการเปรียบเทียบค่า ณ.ตำแหน่ง Index นั้นๆ ของ Pointer A ,B ดังแสดงได้ดังภาพที่ 4.30 , ภาพที่ 4.31 , ภาพที่ 4.32

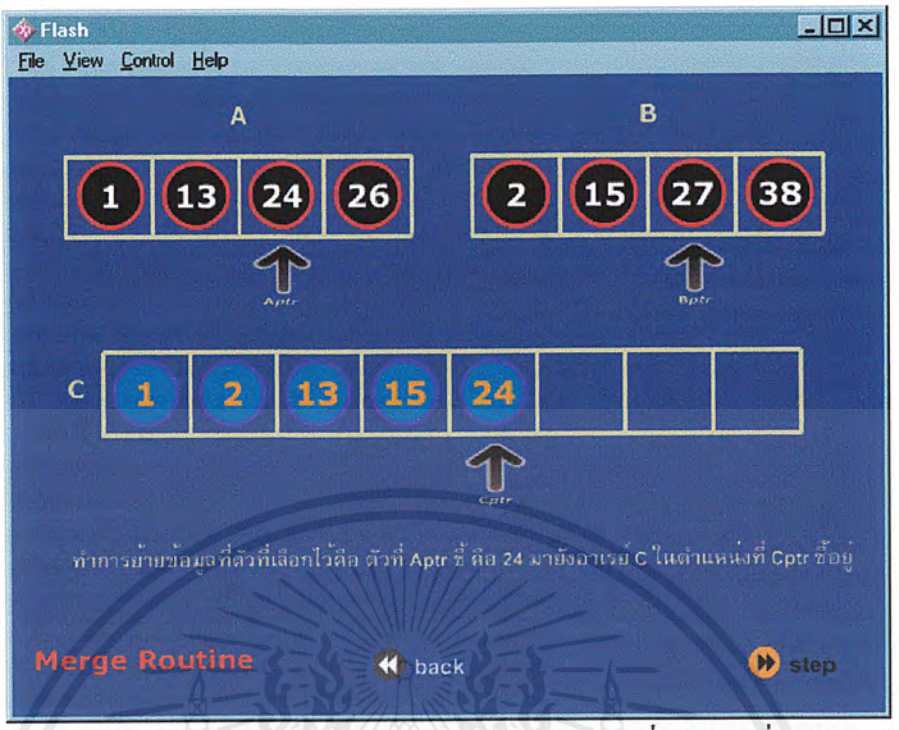


ภาพที่ 4.30 แสดงหน้าจอการทำงานของโปรแกรมสาธิตการ Merge เมื่อเลื่อน Pointer B,Cไปยัง Index ถัดไป

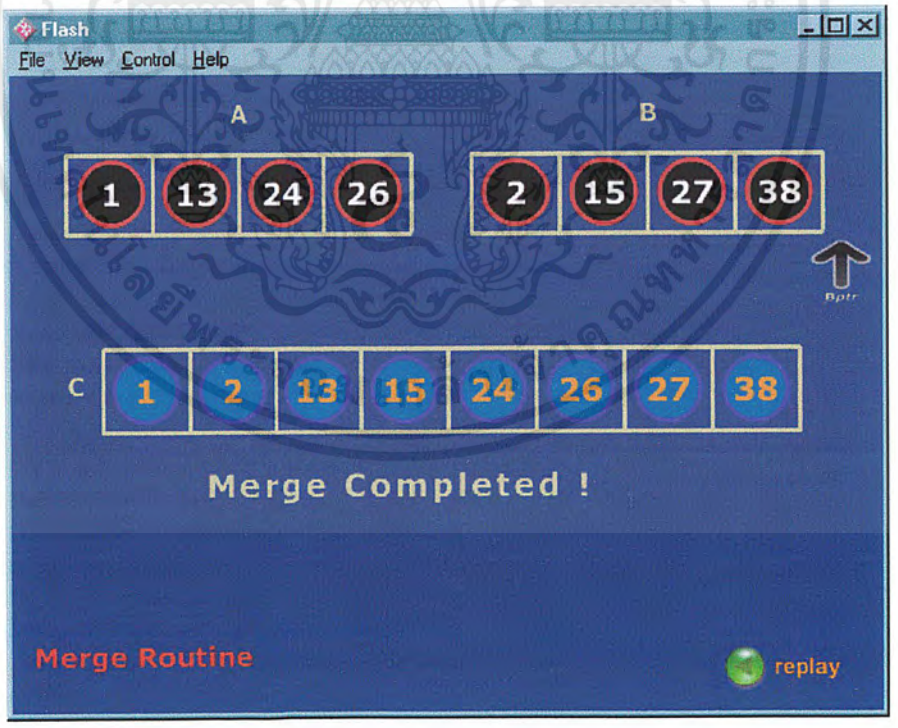


ภาพที่ 4.31 แสดงหน้าจอการทำงานของโปรแกรมสาธิตการ Merge เมื่อทำการเปรียบเทียบค่าณ.ตำแหน่งPointer Aptr,Bptr ทำการเลือกค่าที่น้อยกว่าคือค่า ณ. ตำแหน่ง Pointer Aptr มาเก็บในC ณ.ตำแหน่ง Cptr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



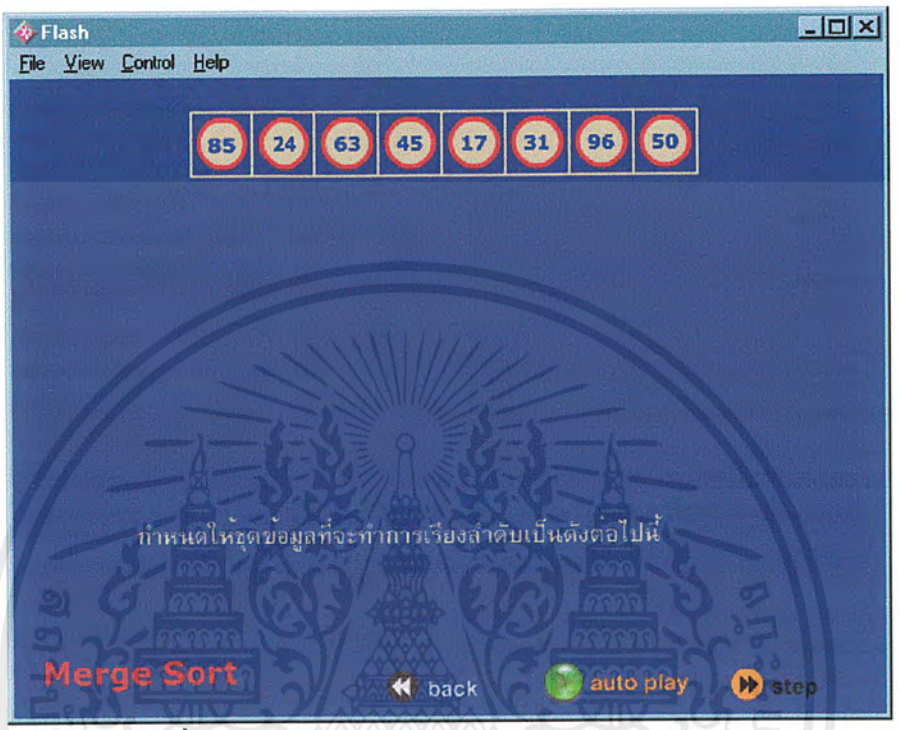
ภาพที่ 4.32 แสดงหน้าจอการทำงานของโปรแกรมสาธิตการ Mergeเมื่อทำการเลื่อนตำแหน่ง Pointer Bptr และ Cptr ไปยังตำแหน่งถัดไป



ภาพที่ 4.33 แสดงหน้าจอผลลัพธ์สุดท้ายของโปรแกรมสาธิตการ Merge

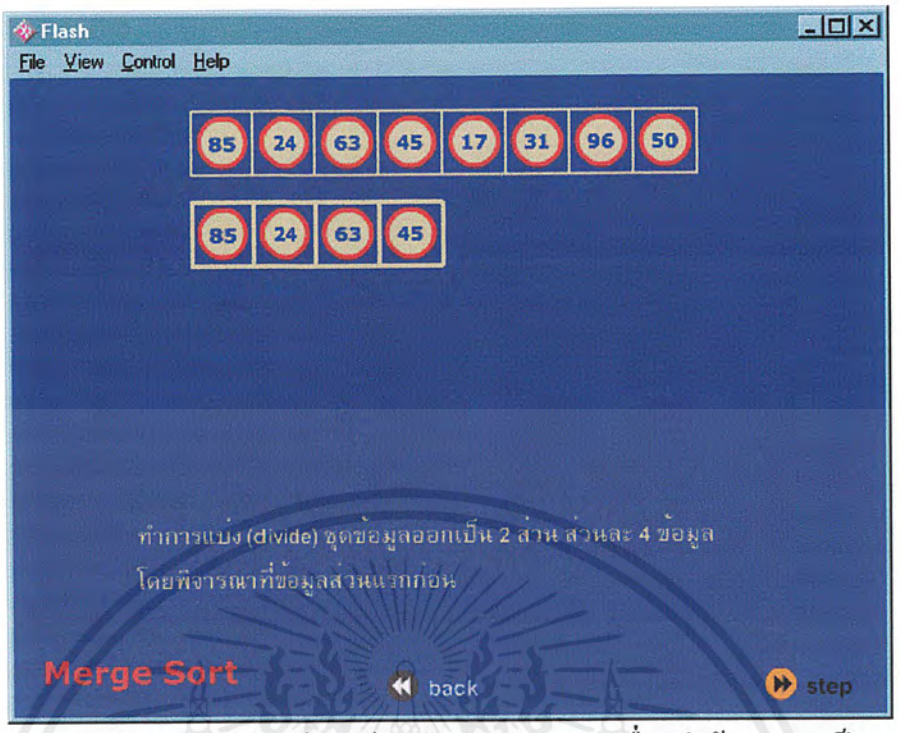
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรมสาธิตการจัดเรียงข้อมูลด้วยวิธี Merge Sort นี้ในขั้นตอนที่สองจะทำการสาธิตในส่วนของขั้นตอนการเรียงลำดับชุดข้อมูลซึ่งประกอบด้วยข้อมูลคือ 85 24 63 45 17 31 96 50 ด้วยวิธี Merge Sort ซึ่งสามารถแสดงได้ด้วยภาพที่ 4.34

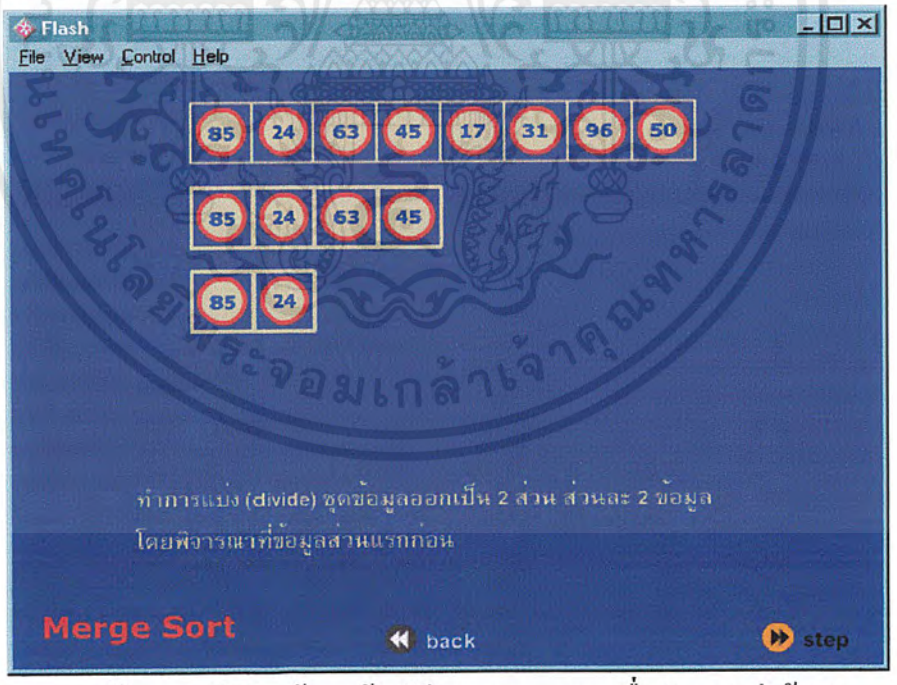


ภาพที่ 4.34 แสดงหน้าจอเริ่มต้นของการจัดเรียงข้อมูลด้วยวิธีการ Merge Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

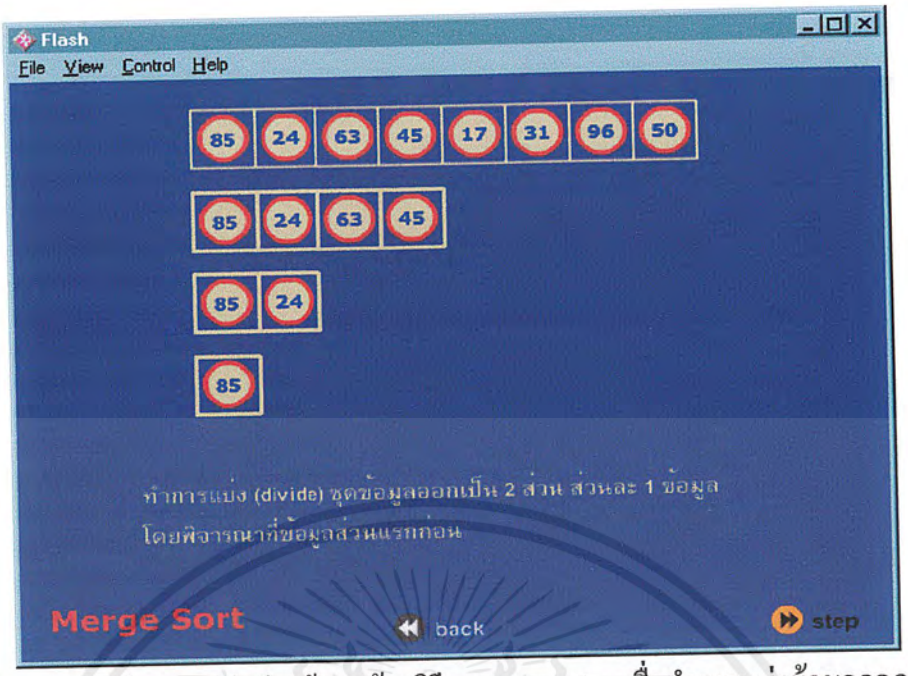


ภาพที่ 4.35 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อแบ่งข้อมูลออกเป็นสองส่วนและดำเนินการ Merge Sort กับข้อมูลในแต่ละส่วน

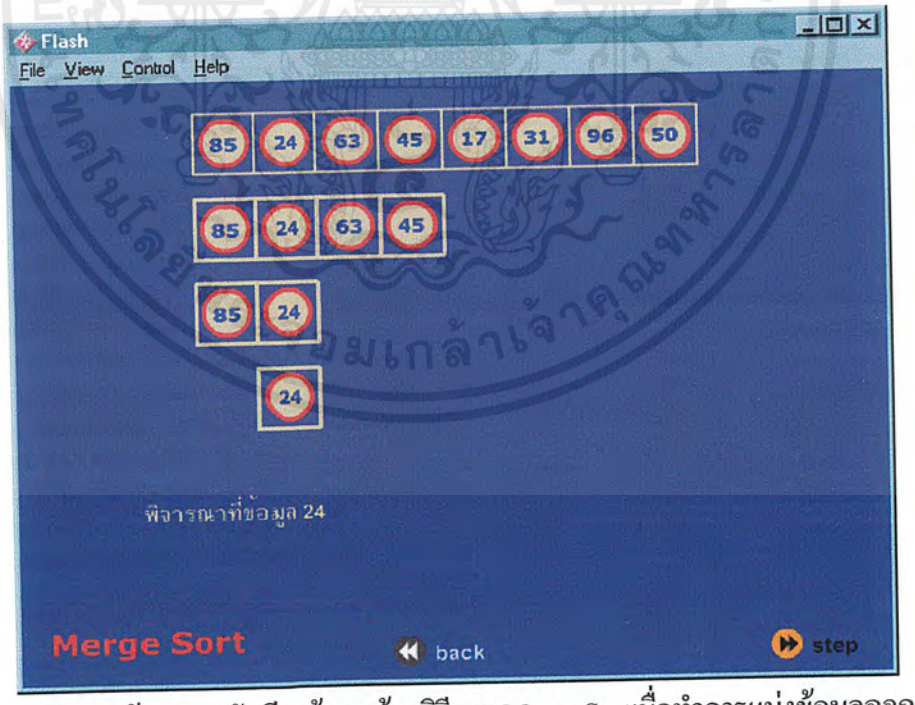


ภาพที่ 4.36 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อทำการแบ่งข้อมูลออกเป็นสองส่วนส่วนละเท่าๆกันดำเนินการ Merge Sort กับข้อมูลในแต่ละส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



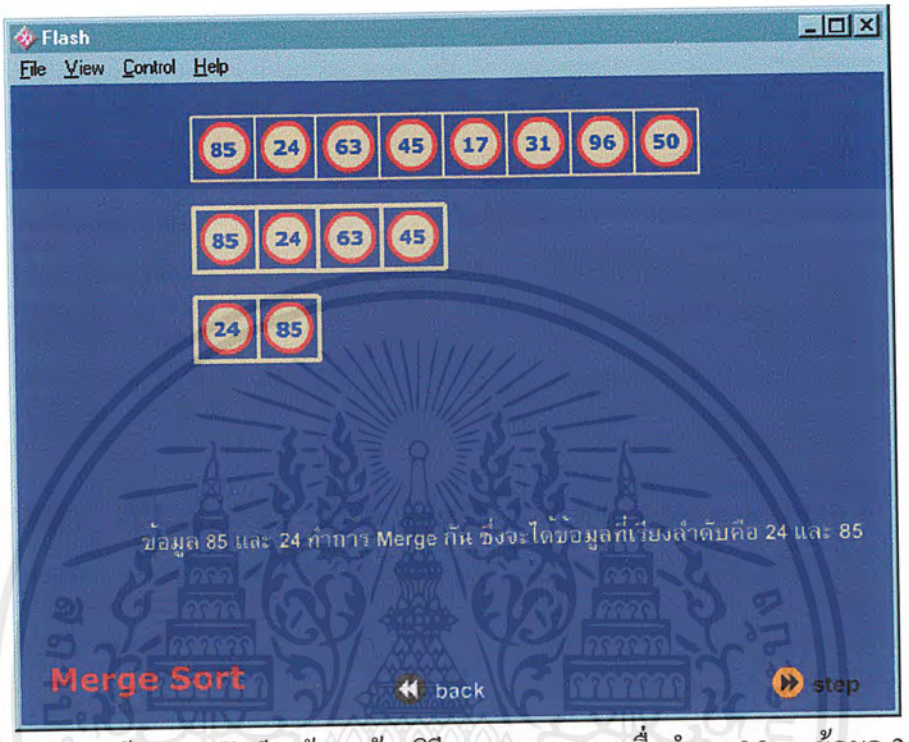
ภาพที่ 4.37 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อทำการแบ่งข้อมูลออกเป็นสองส่วนส่วนละเท่าๆกันดำเนินการ Merge Sort กับข้อมูลในแต่ละส่วนและเมื่อข้อมูลเหลือเพียง 1 ตัวไม่สามารถทำการแบ่งได้แล้วให้ดำเนินการ Merge Sort กับชุดข้อมูลในส่วนที่เหลือ



ภาพที่ 4.38 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อทำการแบ่งข้อมูลออกเป็นสองส่วนส่วนละเท่าๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

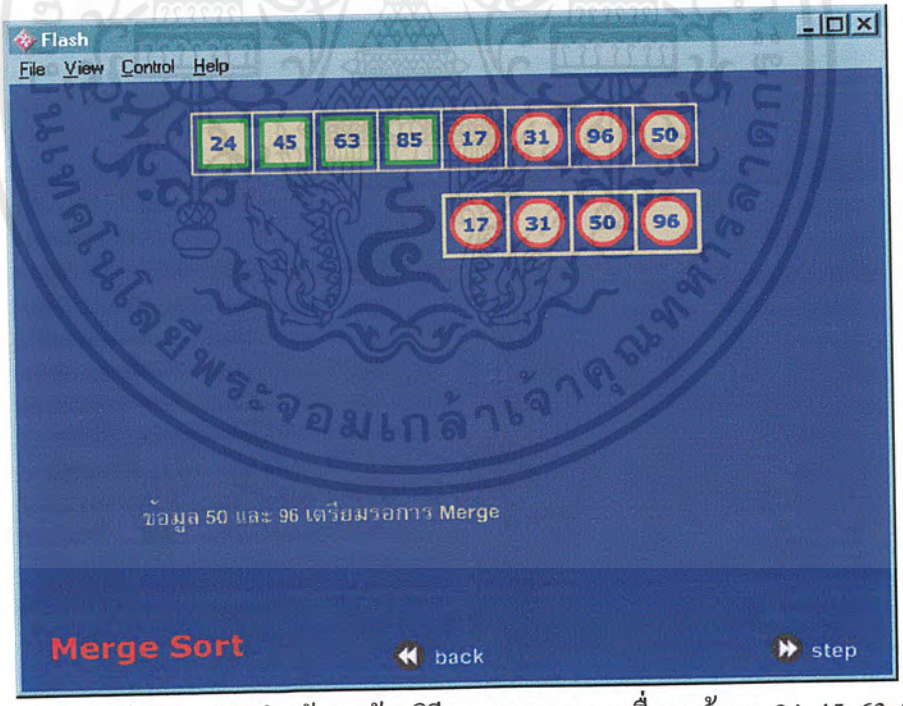
จากภาพที่ 4.38 ทำการแบ่งข้อมูลออกเป็นสองส่วนส่วนละเท่าๆกันดำเนินการ Merge Sort กับข้อมูลในแต่ละส่วน และเมื่อข้อมูลเหลือเพียง 1 ตัวไม่สามารถทำการแบ่งได้แล้วให้ดำเนินการ Merge Sort กับชุดข้อมูลในส่วนที่เหลือพิจารณา 24



ภาพที่ 4.39 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อทำการMergeข้อมูล 24 85 ด้วยวิธีการ Merge ข้างต้น จากนั้นพิจารณาทำการ Merge Sort กับชุดข้อมูลที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

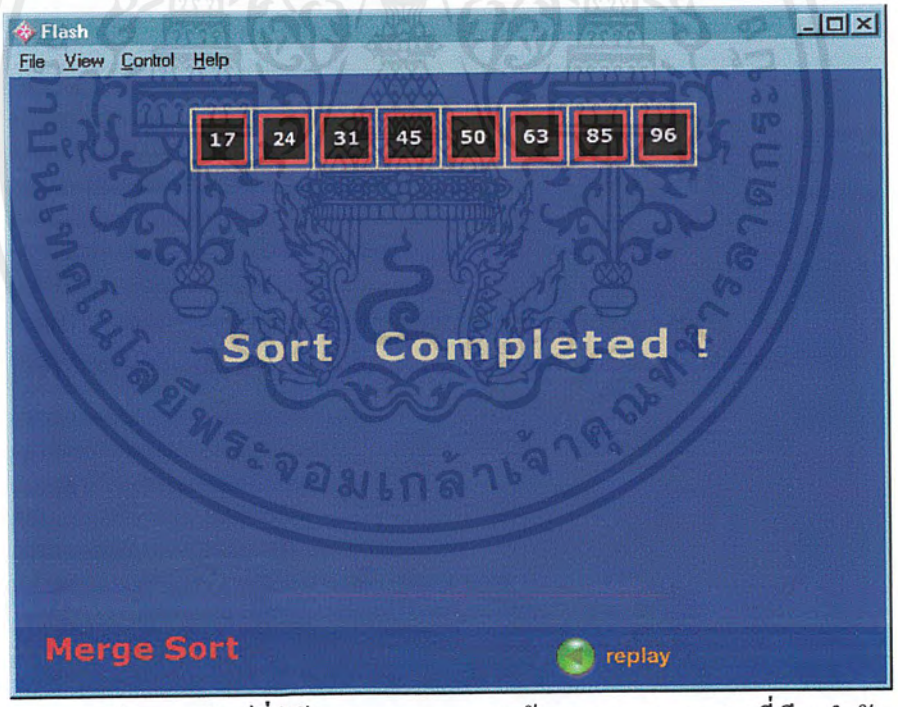
ภาพที่ 4.40 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อทำการ MergeSort ชุดข้อมูล 63 45 และดำเนินการเช่นเดิมจนกระทั่งได้ชุดข้อมูลดังภาพที่ 4.41



ภาพที่ 4.41 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อชุดข้อมูล 24 45 63 85 ที่เรียงลำดับแล้ว รอการ Merge กับชุดข้อมูล 17 31 50 96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.42 แสดงหน้าจอการจัดเรียงข้อมูลด้วยวิธีการ Merge Sortเมื่อชุดข้อมูล 24 45 63 85 ที่เรียงลำดับแล้ว รอการ Merge กับชุดข้อมูล 17 31 50 96 ที่ทำการเรียงลำดับแล้ว

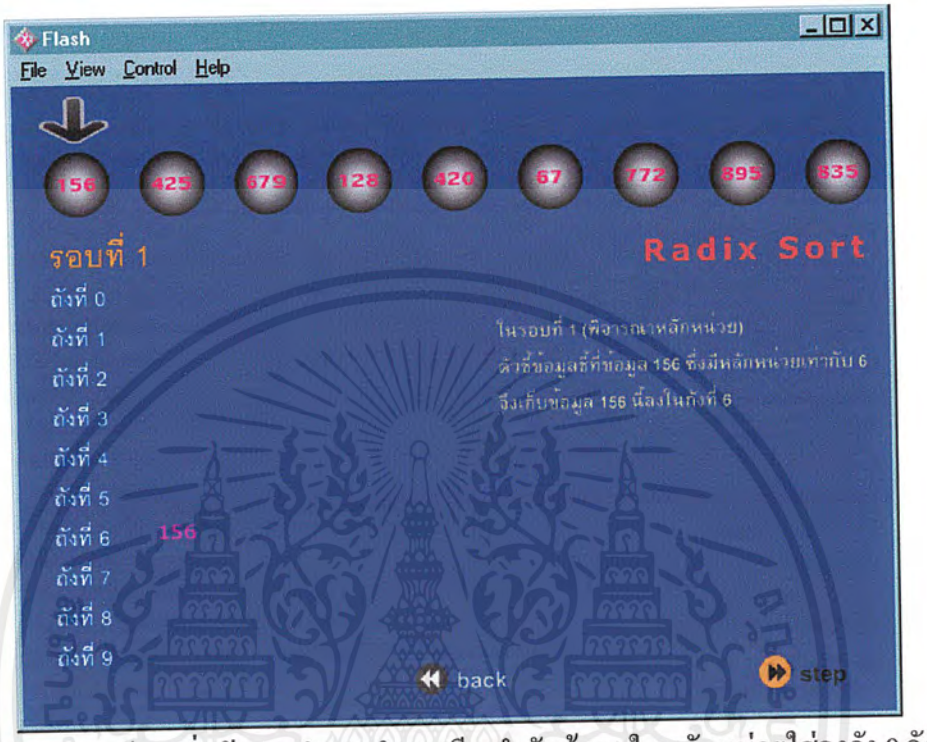


ภาพที่ 4.43 แสดงหน้าจอผลลัพธ์ที่ ได้จากการ Merge ชุดข้อมูล 24 45 63 85 ที่เรียงลำดับแล้วกับชุดข้อมูล 17 31 50 96 ที่ทำการเรียงลำดับแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Radix Sort

โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Radix Sort สามารถอ้างอิงได้ด้วยภาพที่ 2.26 ในหัวข้อ Radix Sort โดยชุดตัวอย่างข้อมูลมีจำนวน 9 ตัว มีข้อมูลคือ 156 425 679 128 420 67 772 895 835



ภาพที่ 4.44 แสดงหน้าจอเริ่มต้นการทำงานทำการเรียงลำดับข้อมูลในหลักหน่วยใส่ลงถัง 9 ถัง

ทำการแยกข้อมูลทั้ง 9 ตัวลงในถังทั้ง 9 ถังตามหลักหน่วยจะได้ผลลัพธ์ดังภาพที่ 4.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.45 แสดงหน้าจอผลลัพธ์ จากการทำงานในรอบที่ 1 นี้ คือ 420 772 425 895 835 156 67 128 679 เพื่อไปใช้ในการทำงานในรอบที่สองต่อไป

ภาพที่ 4.46 แสดงหน้าจอเมื่อทำการเรียงข้อมูลผลลัพธ์ที่ได้จากรอบที่1ทั้ง 9 ตัวลงในถังให้ครบทั้ง 9 ถัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 4.46 เมื่อทำการเรียงข้อมูลผลลัพธ์ที่ได้จากรอบที่1ทั้ง 9 ตัวลงในถังให้ครบทั้ง 9 ถัง โดยพิจารณาหลักสิบจะได้ผลลัพธ์การทำงานในรอบที่2คือ 420 425 128 835 156 067 772 679 895 เพื่อไปใช้ในการทำงานในรอบที่สามต่อไป

เมื่อเรียงข้อมูลทั้ง 9 ตัวลงในถังให้ครบทั้ง 9 ถังโดยพิจารณาหลักร้อยจะได้ผลลัพธ์การทำงานในรอบ3คือ 067 128 156 420 425 679 772 835 895 ซึ่งเป็นผลลัพธ์ที่ได้จากการเรียงลำดับด้วยวิธี Radix Sort และจะปรากฏหน้าจอผลลัพธ์ดังภาพที่ 4.47

ภาพที่ 4.47 แสดงหน้าจอสิ้นสุดการจัดเรียงข้อมูลด้วย Radix Sort

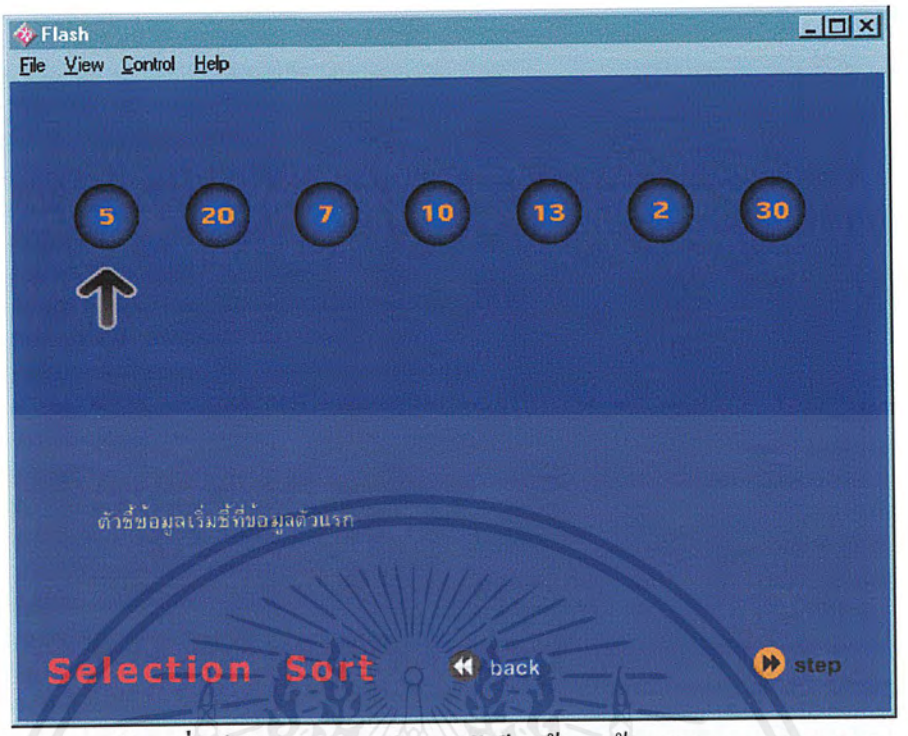
4.1.5 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort

จากหลักการทำงานด้วยวิธีนี้คือ(ถ้าชุดของข้อมูลเป็นลิสต์ ประกอบด้วย a[1] , a[2] , . . . , a[n])

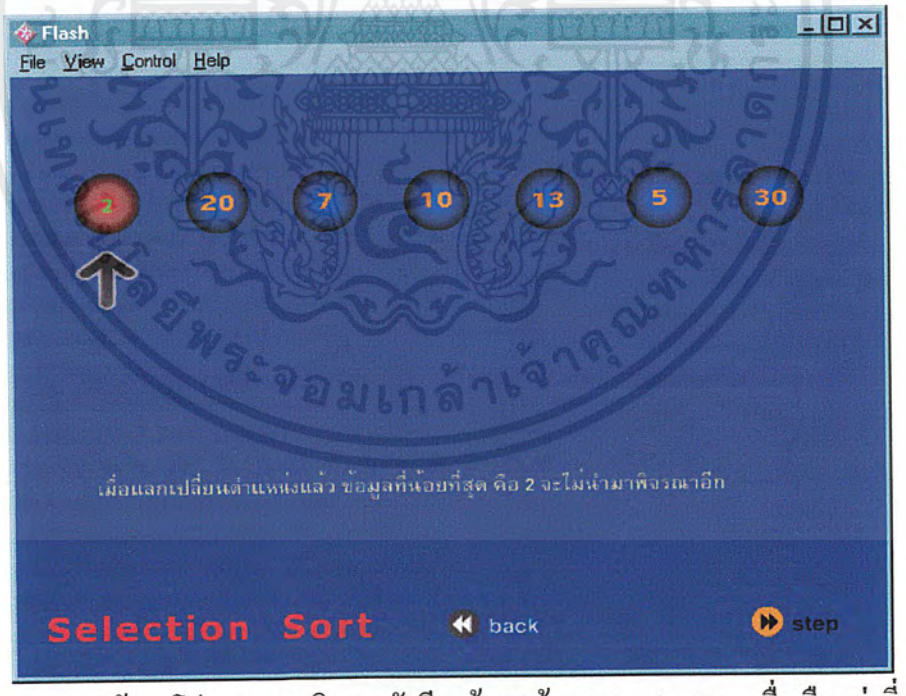
1. เลือกค่าที่น้อยที่สุด
2. นำมาแลกเปลี่ยนลำดับกับค่าใน a[1]

หลังจากนั้นก็ทำตามหลักการทั้ง 2 กับข้อมูลที่เหลือ คือในครั้งที่ 2 ค่า a[2] ถูกแลกเปลี่ยนกับค่าที่เลือกแล้วว่าน้อยที่สุดในลิสต์ a[2]...a[n] และในครั้งที่ 3 ค่า a[3] ถูกแลกเปลี่ยนกับค่าที่น้อยที่สุดใน a[3]...a[n] และเรื่อยไปจนเหลือสมาชิกในลิสต์สองค่าคือ a[n-1] และ a[n] ดังนั้นจำนวนรอบในการกระทำเป็น n-1 รอบสามารถแสดงโปรแกรมสาธิตได้ดังภาพที่ 4.48, 4.49, 4.50 ,4.51และ4.52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.48 แสดงหน้าจอเริ่มต้นโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort Pointer ชี้ที่ Index a[1]



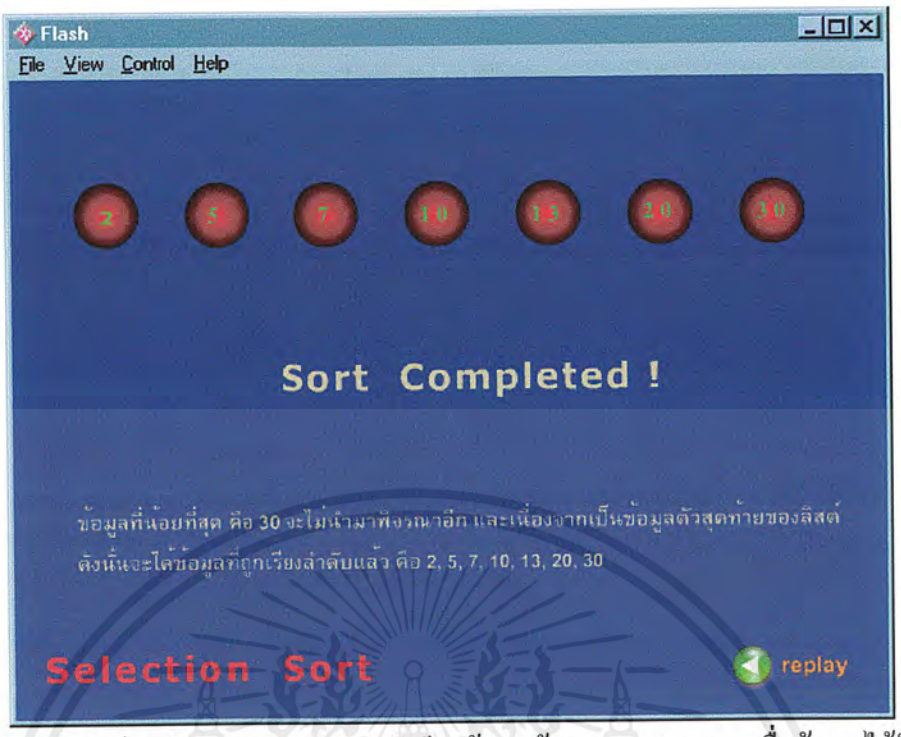
ภาพที่ 4.49 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort เมื่อเลือกค่าที่น้อยที่สุด คือ 2 แล้วเปลี่ยนกับ a[1] และดำเนินการเช่นนี้ ในรอบที่ 2 และรอบต่อๆไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.50 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort เมื่อทำการเลื่อน Index ไปยังตำแหน่งถัดไปและเลือกค่าที่น้อยที่สุดมาแลกเปลี่ยนกับค่าในตำแหน่งของ Index

ภาพที่ 4.51 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort เมื่อข้อมูลที่น้อยที่สุดคือ 5 นำมาแลกเปลี่ยนกับค่าใน a[2] แล้วเลื่อนตัวชี้ไปยังตำแหน่งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.52 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Selection Sort เมื่อข้อมูลได้รับการเรียงลำดับเรียบร้อยแล้ว

4.1.6 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Heap Sort

จากหลักการทำงานของการจัดเรียงข้อมูลด้วยวิธี Heap Sort ซึ่งแบ่งเป็น 2 ขั้นตอนคือ

ขั้นตอนที่ 1

1. รับข้อมูลเข้าสร้างเป็นต้นไม้
2. ปรับต้นไม้ให้มีคุณสมบัติเป็นต้นไม้ฮีปซึ่งหมายถึง $h_i > h_{2i}$ และ $hi > h_{2i+1}$ เมื่อ i=1,2,..n/2

ขั้นตอนที่ 2 นำต้นไม้ฮีปที่ได้จากขั้นตอนที่ 1 มาทำตามอัลกอริทึมนี้

1. r <- - n (* r เป็นตัวแปรแสดงจำนวนข้อมูลในต้นไม้โดยครั้งแรกเซตเป็น n *)
2. ในระหว่างที่ r > 1 ทำ
   2.1 สับเปลี่ยนค่าใน h_1 และ h_r
   2.2 r <- - r — 1 (* ลดจำนวนข้อมูลลง 1 *)
   2.3 ปรับต้นไม้ที่เหลือ h_1 , h_2 ..., h_r ให้เป็นฮีป (* โดยปรับที่เส้นจากเริ่มจาก h_1 เท่านั้น เนื่องจาก h_2 , h_3, ... h_r มีคุณสมบัติเป็นฮีปอยู่แล้ว *)

โดยในขั้นแรกจะทำการสาธิตวิธีการสร้างต้นไม้ฮีปจากชุดข้อมูลสามารถแสดงได้ดังภาพที่ 4.53

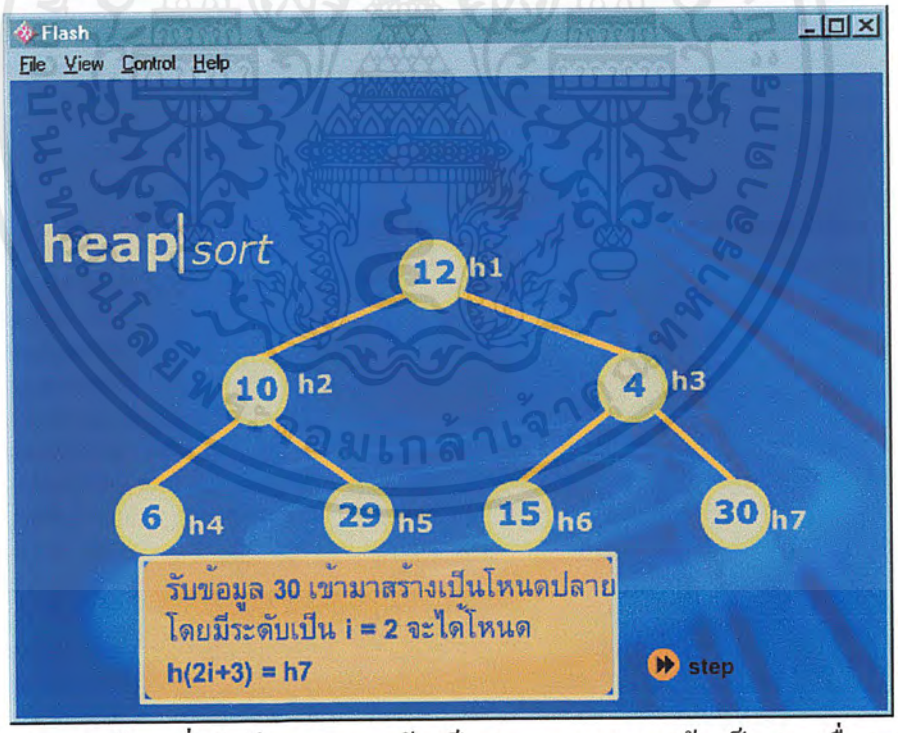
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.53 แสดงหน้าจอเริ่มต้นการสร้างต้นไม้ฮีปจากข้อมูล 12 10 4 6 29 15 30

ภาพที่ 4.54 แสดงหน้าจอเมื่อรับข้อมูล 12 มาสร้างเป็น h_i และ 10 มาสร้างเป็น h_{2i} และ 4 มาสร้างเป็น h_{2i+1} เมื่อ i=1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

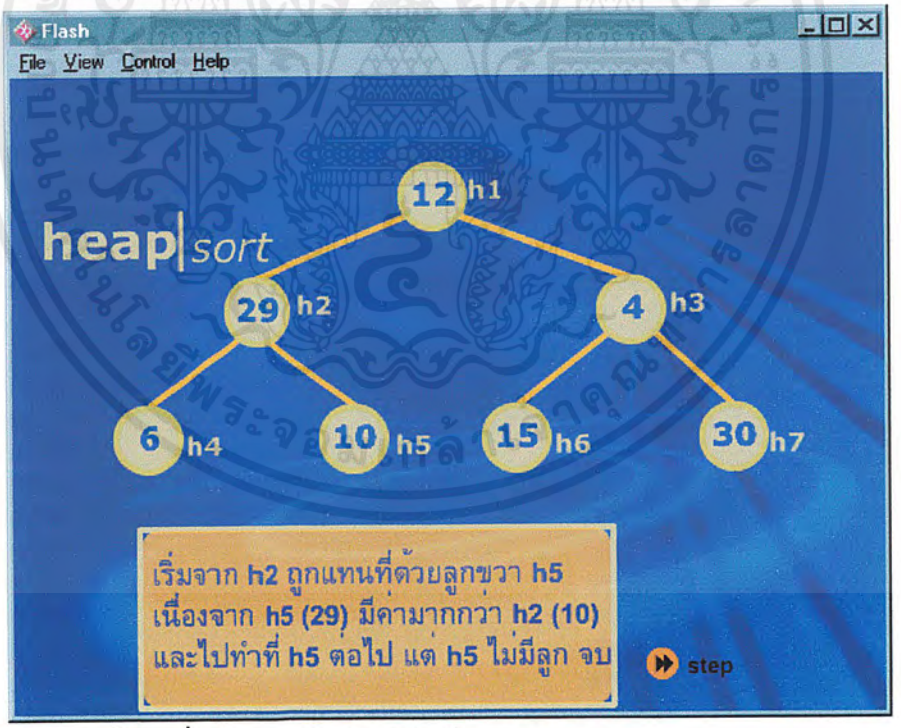
ภาพที่ 4.55 แสดงหน้าจอเมื่อรับข้อมูล 6 มาสร้างเป็น h_{2i} และ 29 มาสร้างเป็น h_{2i+1} เมื่อ i=2



ภาพที่ 4.56 แสดงหน้าจอเมื่อรับข้อมูล 15 มาสร้างเป็น h_{2i+2} และ 30 มาสร้างเป็น h_{2i+3} เมื่อ i=2

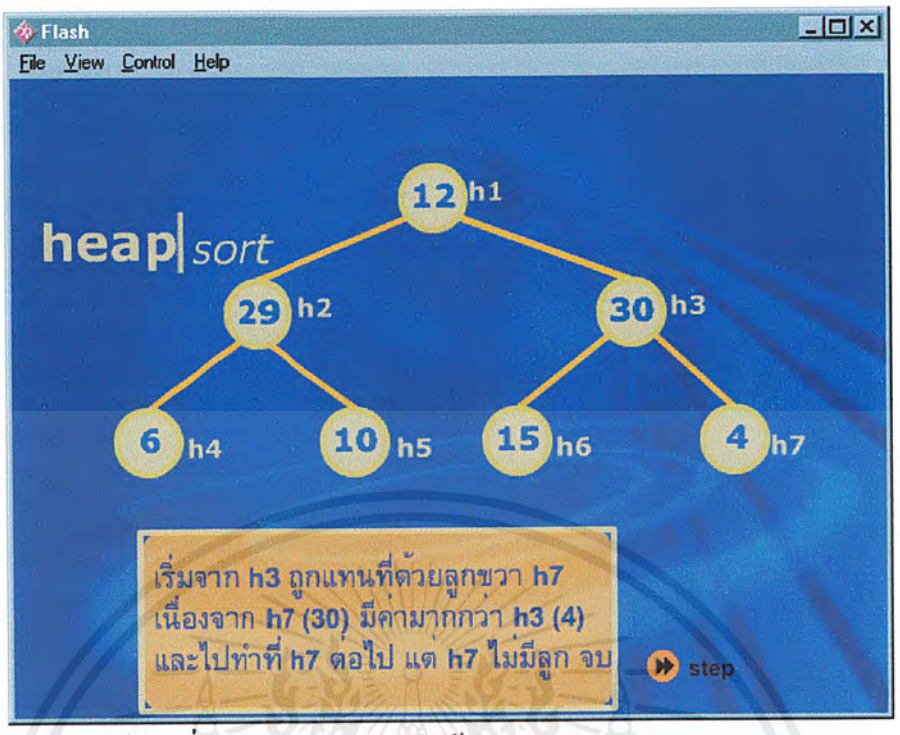
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.57 แสดงหน้าจอเริ่มต้นขั้นตอนการปรับต้นไม้ให้มีคุณสมบัติเป็นฮีป



ภาพที่ 4.58 แสดงหน้าจอเมื่อทำการสลับตำแหน่ง ข้อมูล 10 และ 29

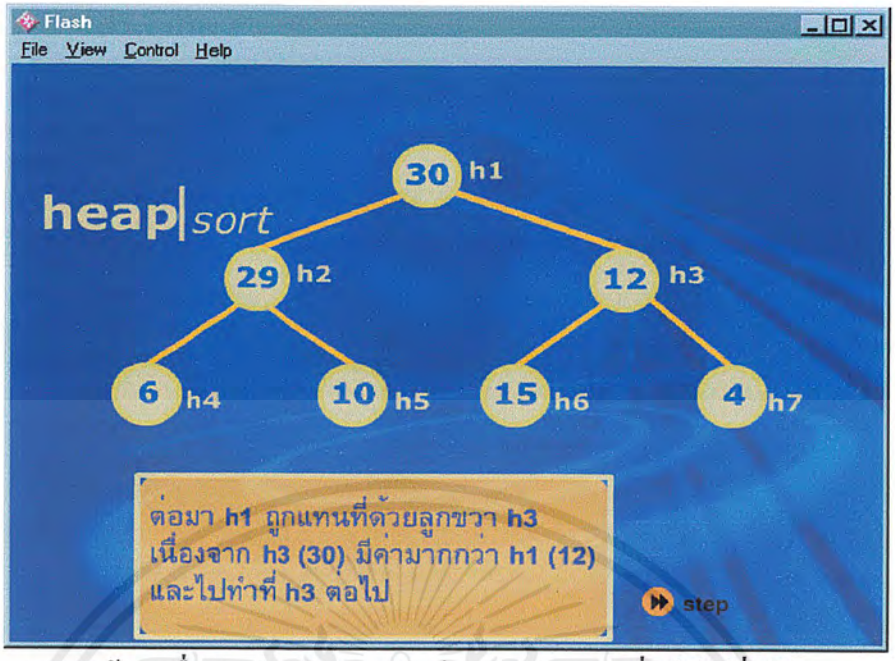
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.59 แสดงหน้าจอเมื่อทำการสลับตำแหน่ง ข้อมูล 4 และ 30

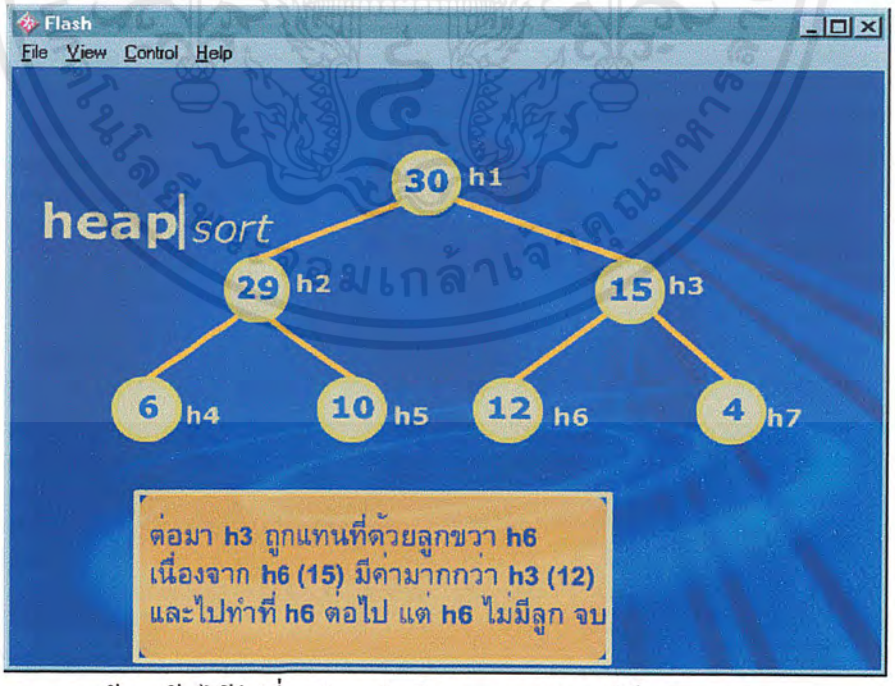
ภาพที่ 4.60 แสดงหน้าจอเมื่อทำการสลับตำแหน่ง ข้อมูล 30 และ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



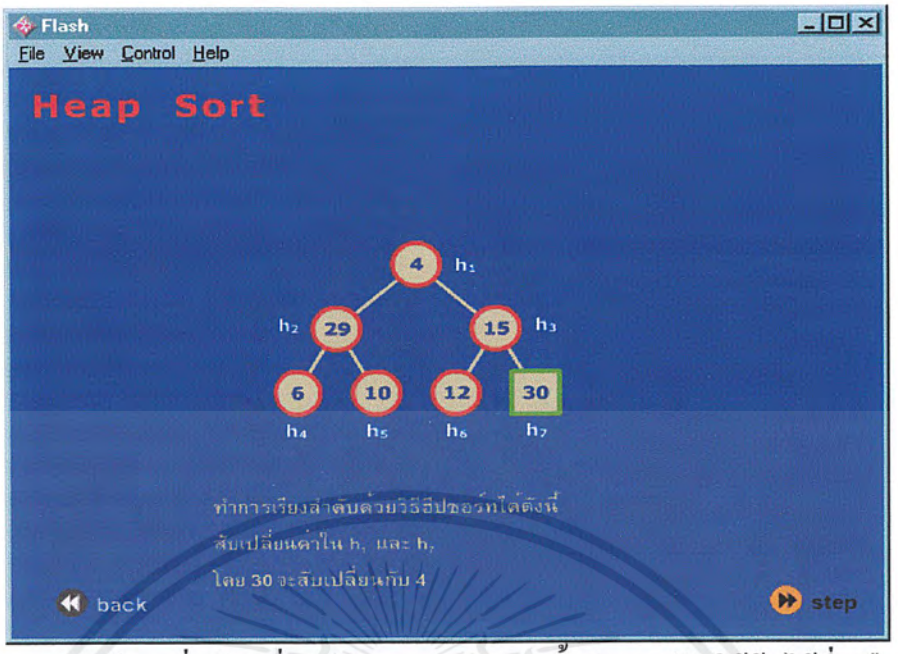
ภาพที่ 4.61 แสดงหน้าจอเมื่อทำการสลับตำแหน่งข้อมูล 12 และ 15 ซึ่งต้นไม้ที่ได้มีคุณสมบัติเป็นฮีปเรียบร้อยแล้ว

โดยในขั้นต่อมาจะทำการสาธิตวิธีการนำต้นไม้ฮีปจากภาพที่ 4.61 มาดำเนินการตามอัลกอริทึม Heap Sort สามารถแสดงได้ดังภาพที่ 4.62

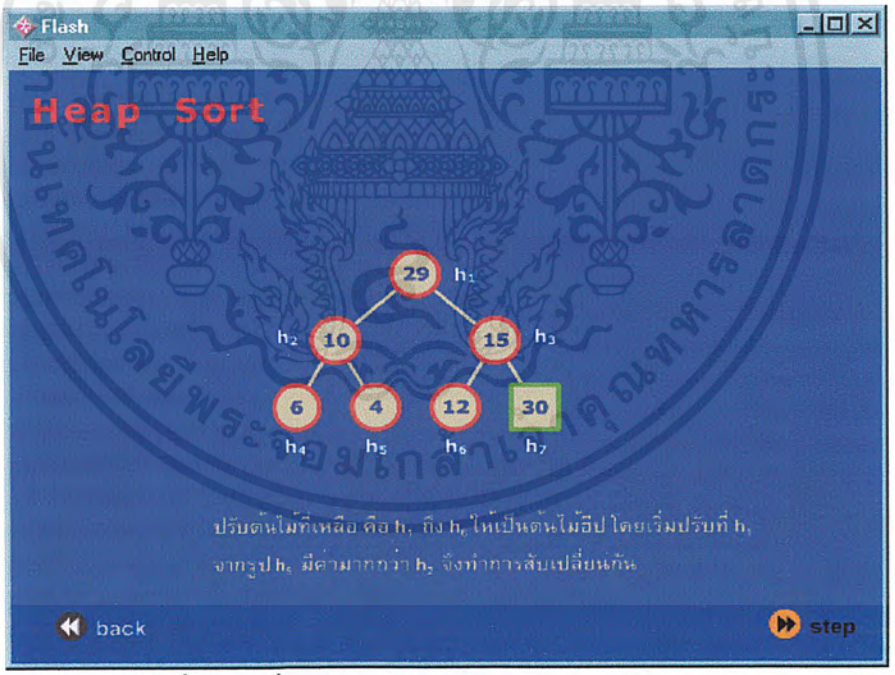


ภาพที่ 4.62 แสดงหน้าจอต้นไม้ฮีปที่จะนำมาดำเนินการตามอัลกอริทึม Heap Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

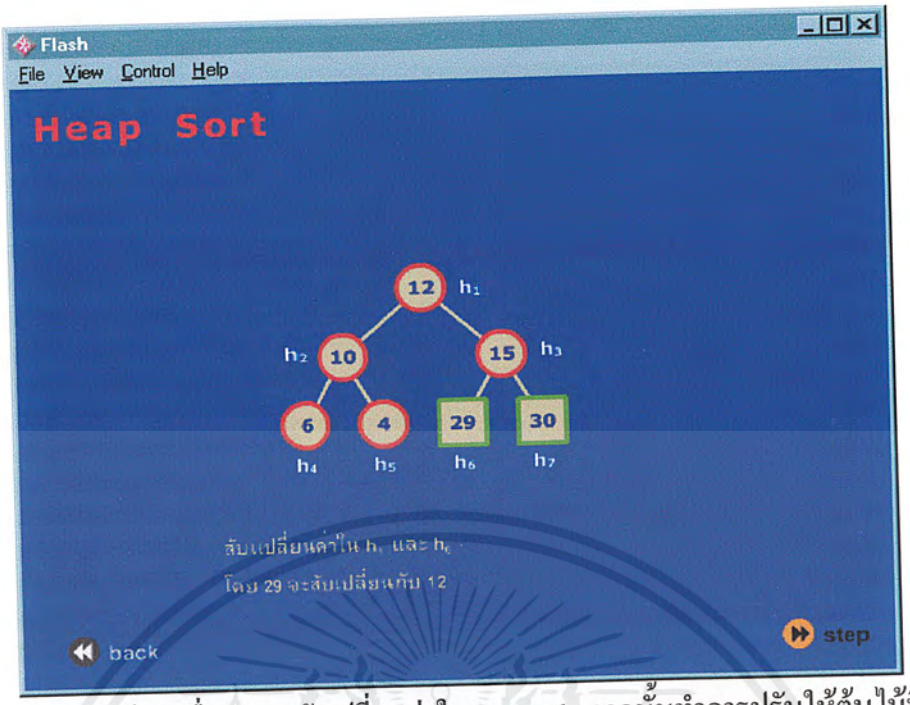


ภาพที่ 4.63 แสดงหน้าจอเมื่อสับเปลี่ยนค่าใน h_l และ h_r จากนั้นทำการปรับให้ต้นไม้ที่เหลือ h_l ถึง h_r มีคุณสมบัติเป็นฮีป

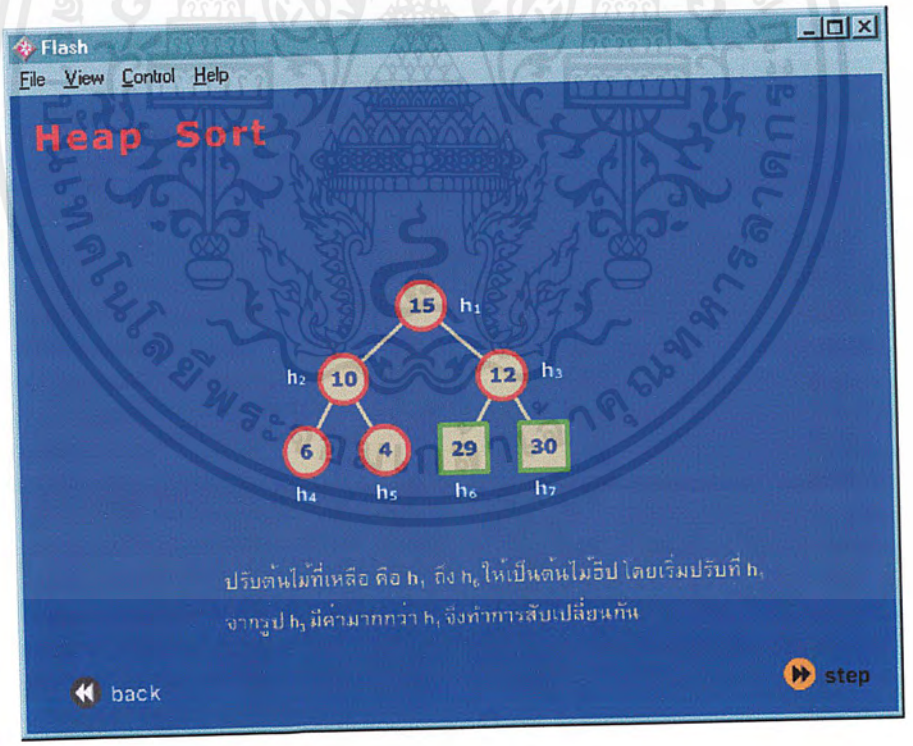


ภาพที่ 4.64 แสดงหน้าจอเมื่อสับเปลี่ยนค่าระหว่าง 4 กับ 29 และ 10 กับ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



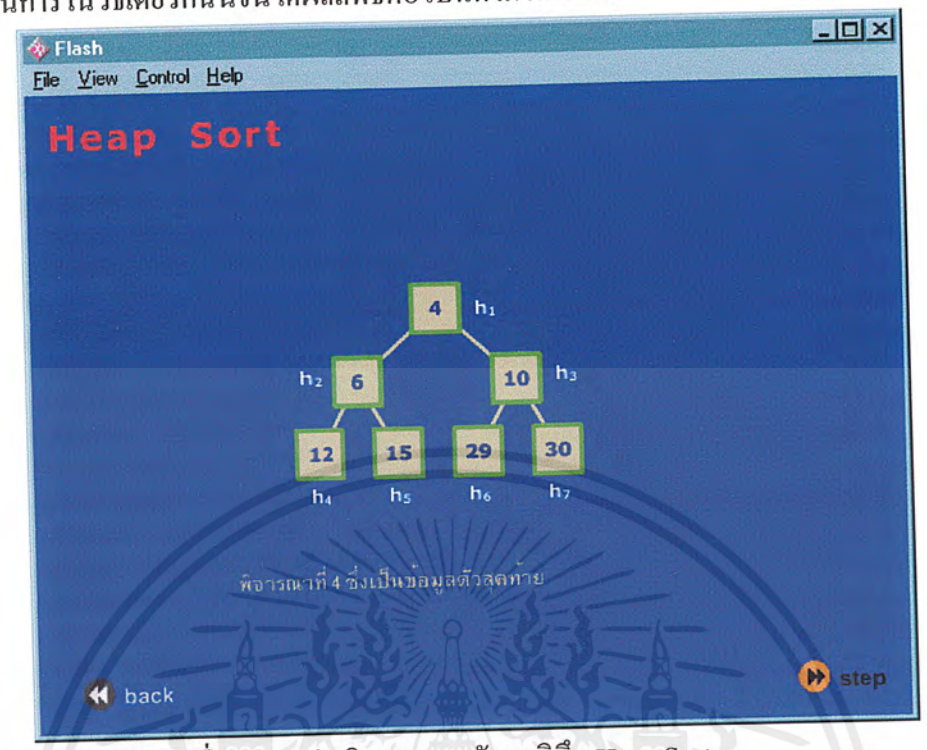
ภาพที่ 4.65 แสดงหน้าจอเมื่อทำการสับเปลี่ยนค่าใน h_1 และ h_6 จากนั้นทำการปรับให้ต้นไม้ที่เหลือ h_1 ถึง h_7มีคุณสมบัติเป็นฮีปได้ดังภาพที่ 4.66



ภาพที่ 4.66 แสดงหน้าจอเมื่อสับเปลี่ยนค่าระหว่าง12 และ 15

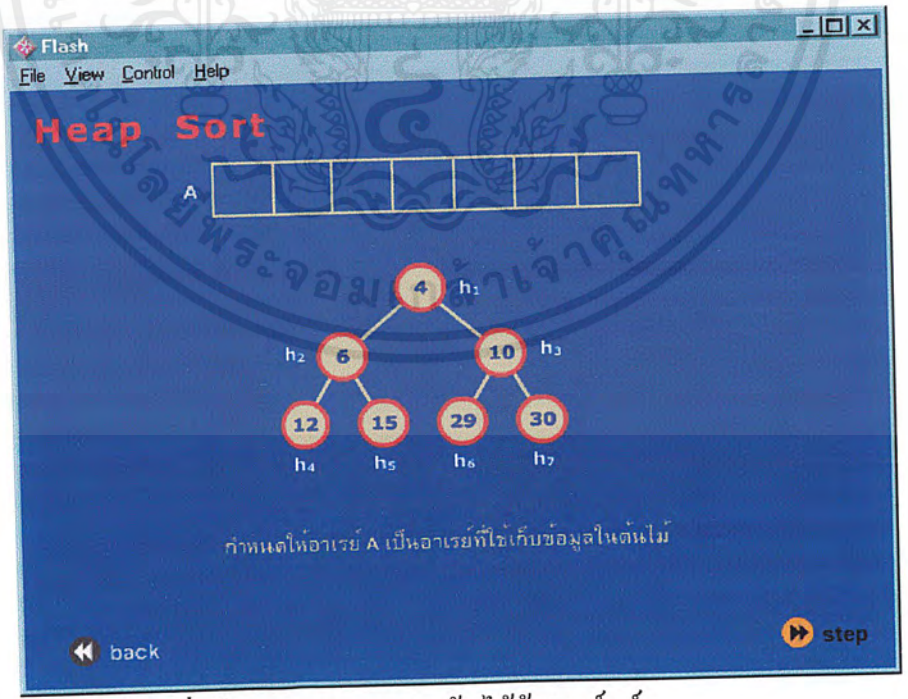
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดำเนินการในวิธีเดียวกันนี้จนได้ผลลัพธ์ต่อไปนี้ดังภาพที่ 4.67



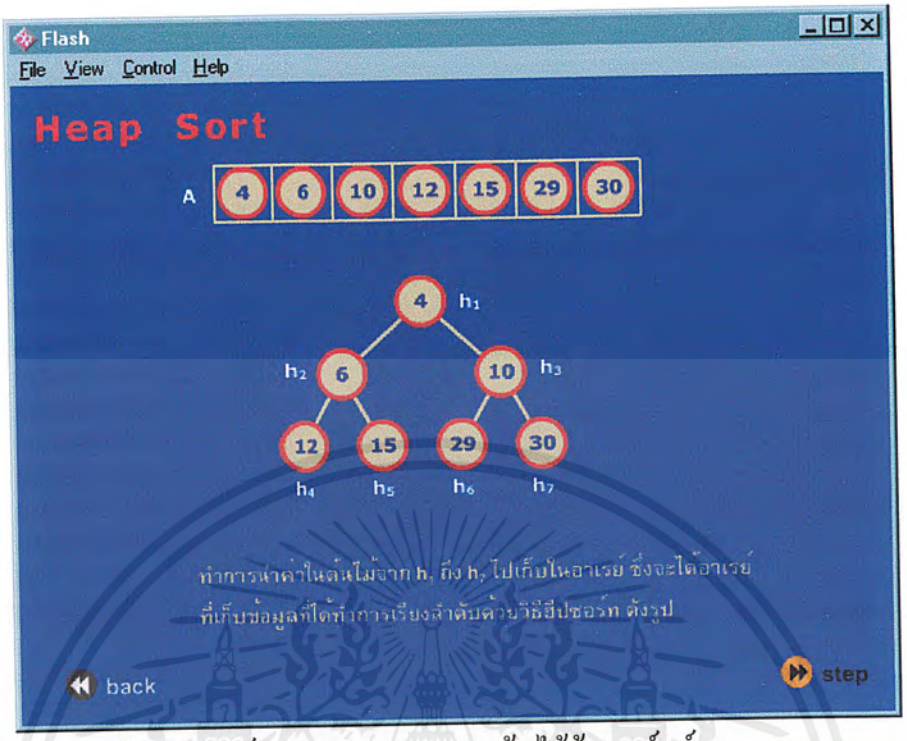
ภาพที่ 4.67 แสดงหน้าจอเมื่อจบการดำเนินการตามอัลกอริทึม Heap Sort

ในขั้นตอนต่อไปจะทำการแทนต้นไม้ด้วยอาร์เรย์ซึ่งมีขั้นตอนในการทำดังภาพที่ 4.68



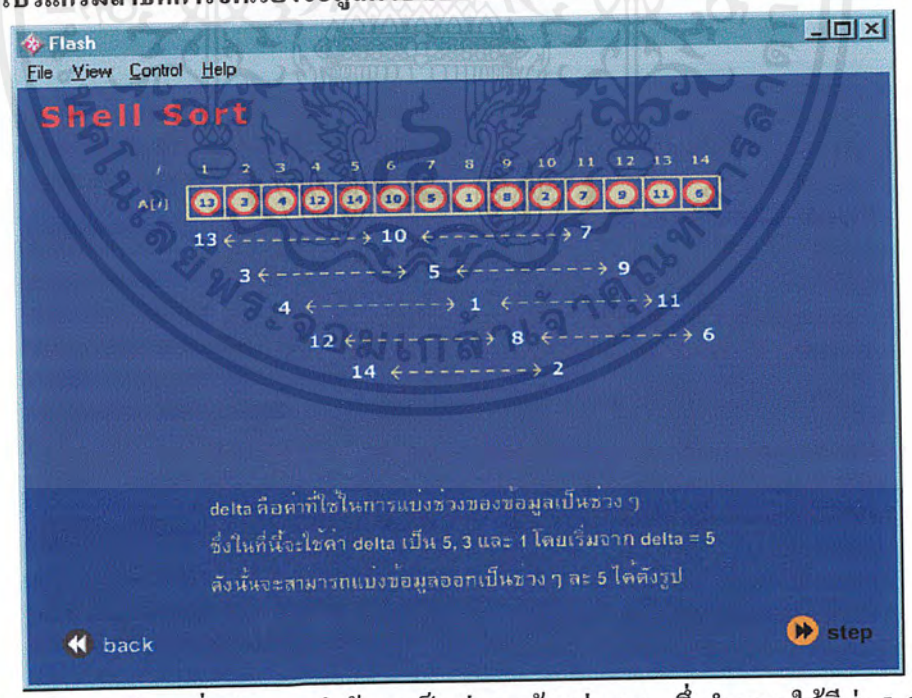
ภาพที่ 4.68 แสดงหน้าจอเริ่มต้นกระบวนการแทนต้นไม้ด้วยอาร์เรย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



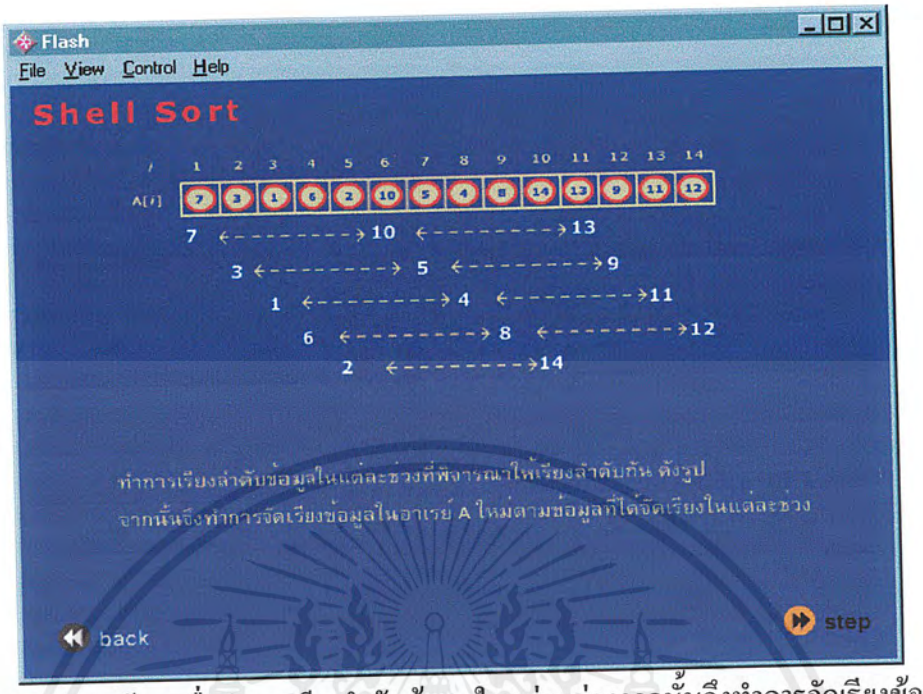
ภาพที่ 4.69 แสดงหน้าจอผลลัพธ์ของกระบวนการแทนต้นไม้ด้วยอาร์เรย์

4.1.7 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Shell Sort

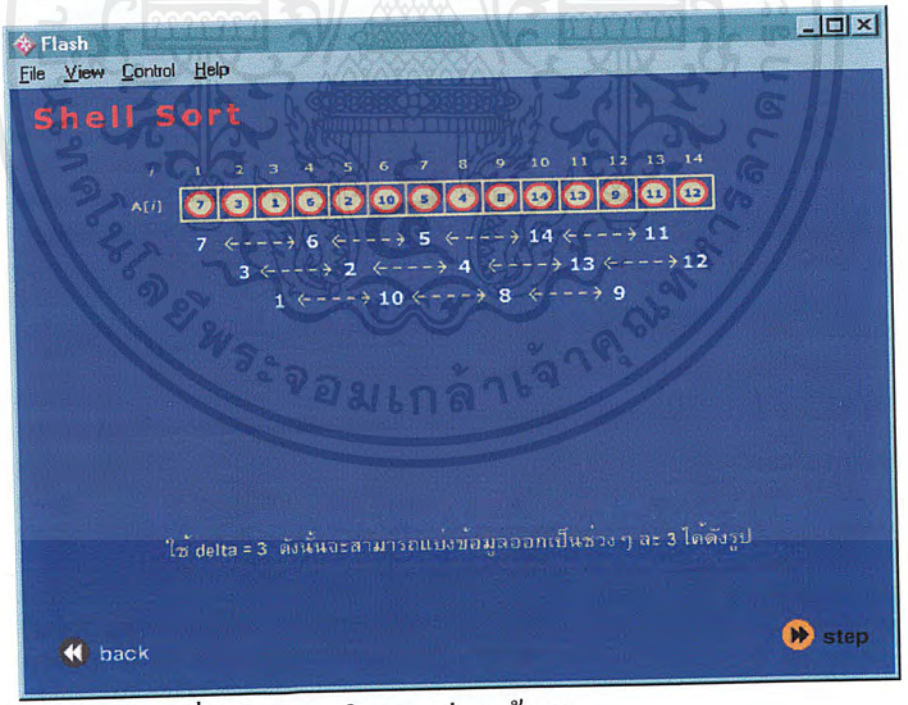


ภาพที่ 4.70 แสดงหน้าจอเมื่อทำการแบ่งข้อมูลเป็นช่วงๆ ด้วยค่า delta ซึ่งกำหนดให้มีค่า 5 3 1 โดยเริ่มจากค่า delta =5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

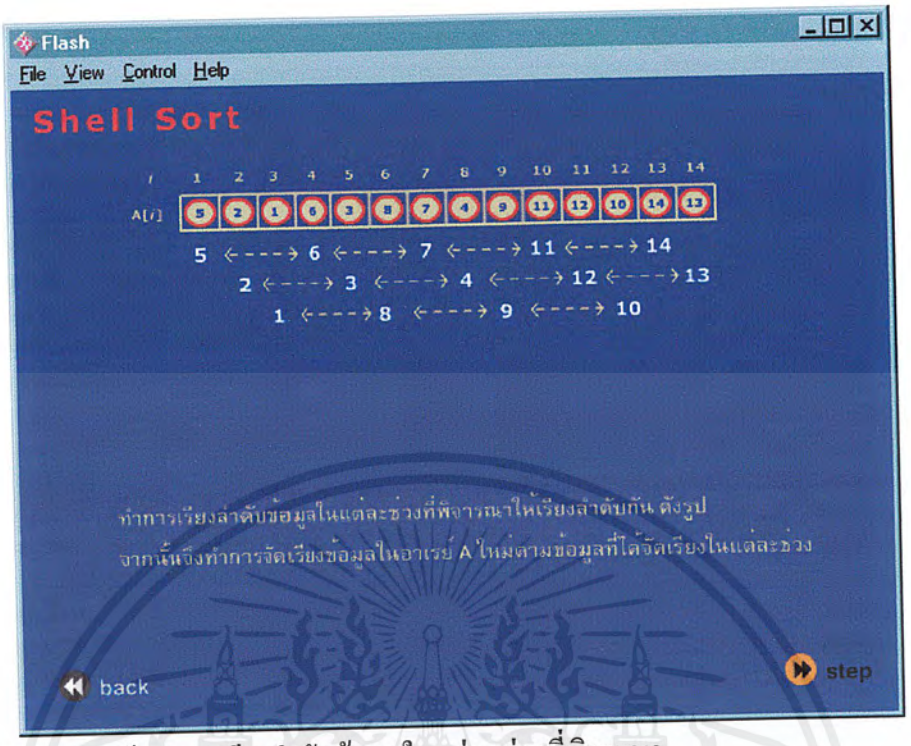


ภาพที่ 4.71 แสดงหน้าจอเมื่อทำการเรียงลำดับข้อมูลในแต่ละช่วงจากนั้นจึงทำการจัดเรียงข้อมูลใน Array A

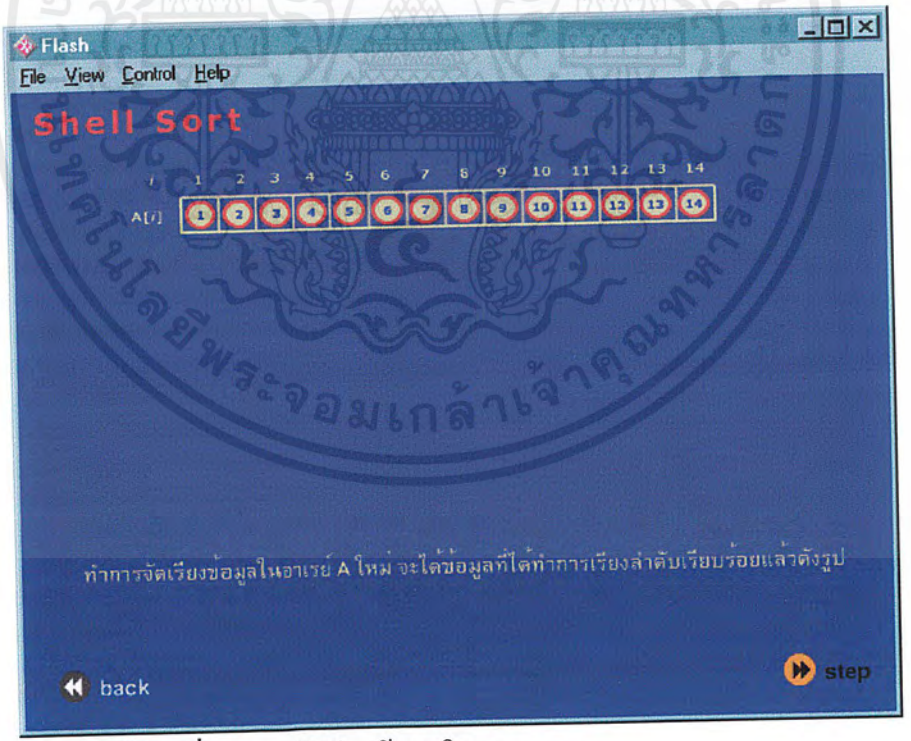


ภาพที่ 4.72 แสดงหน้าจอเมื่อใช้ delta=3 ในการแบ่งชุดข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

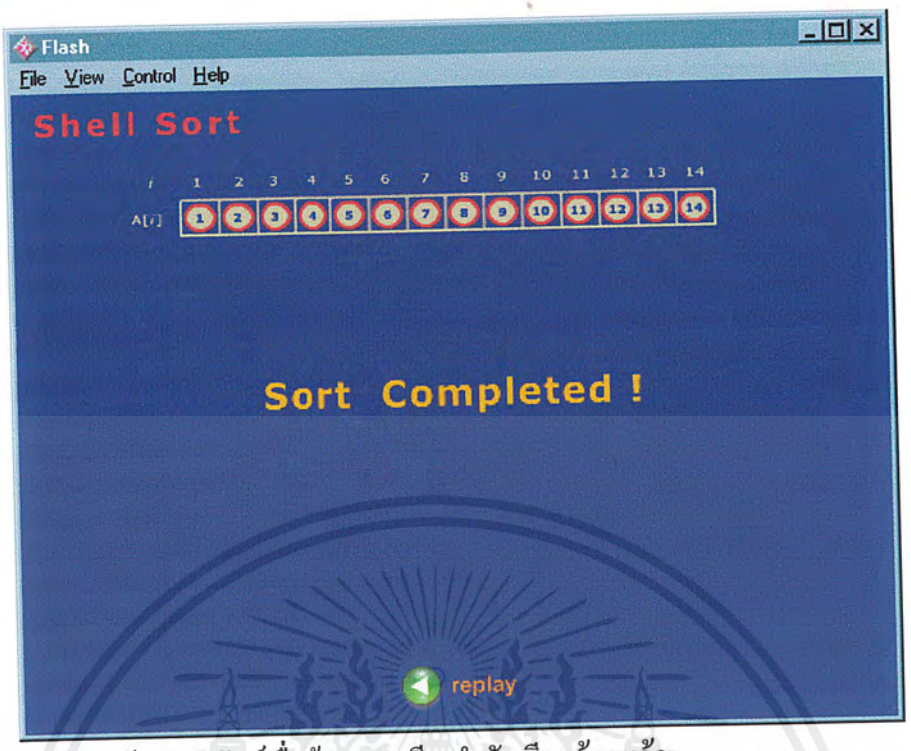


ภาพที่ 4.73 แสดงหน้าจอการเรียงลำดับข้อมูลในแต่ละช่วงที่พิจารณา



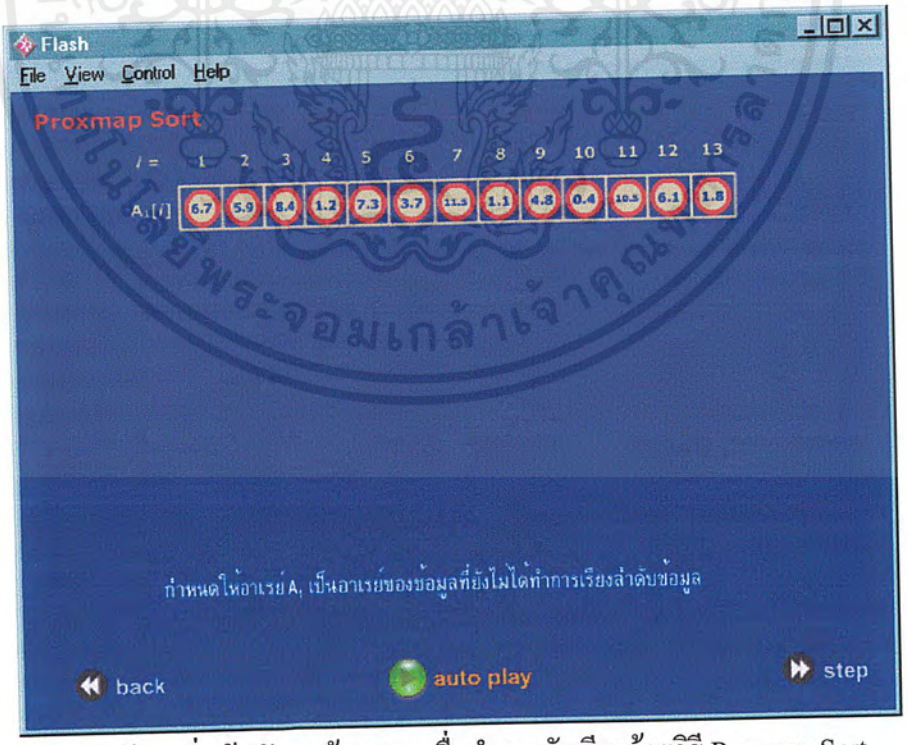
ภาพที่ 4.74 แสดงหน้าจอเมื่อทำการจัดเรียงข้อมูลใน Array A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



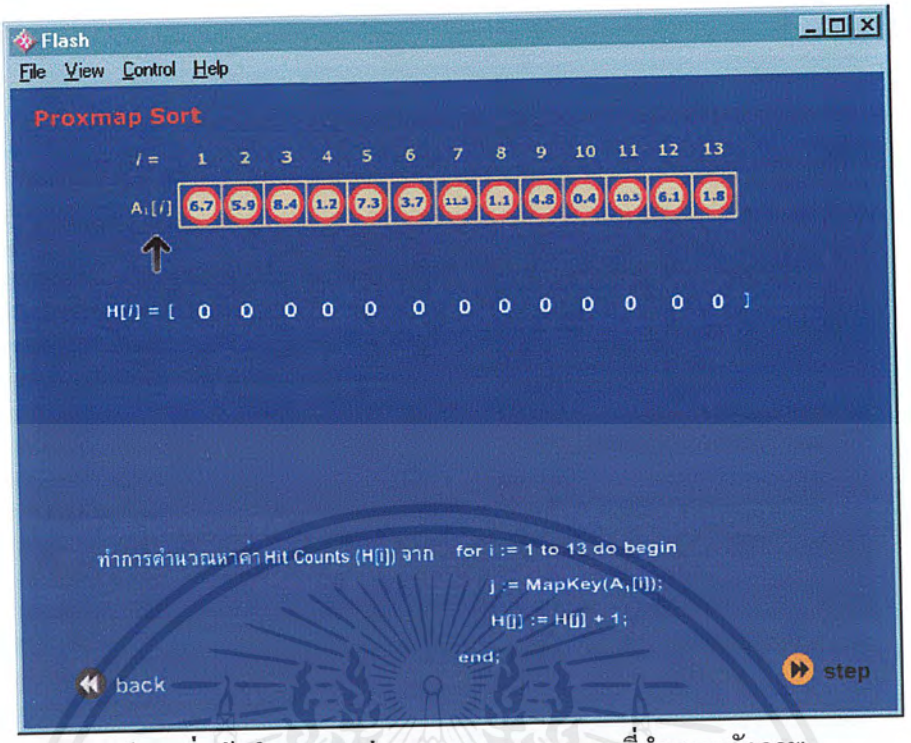
ภาพที่ 4.75 แสดงหน้าจอผลลัพธ์เมื่อข้อมูลถูกเรียงลำดับเรียบร้อยแล้ว

4.1.8 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Proxmap Sort

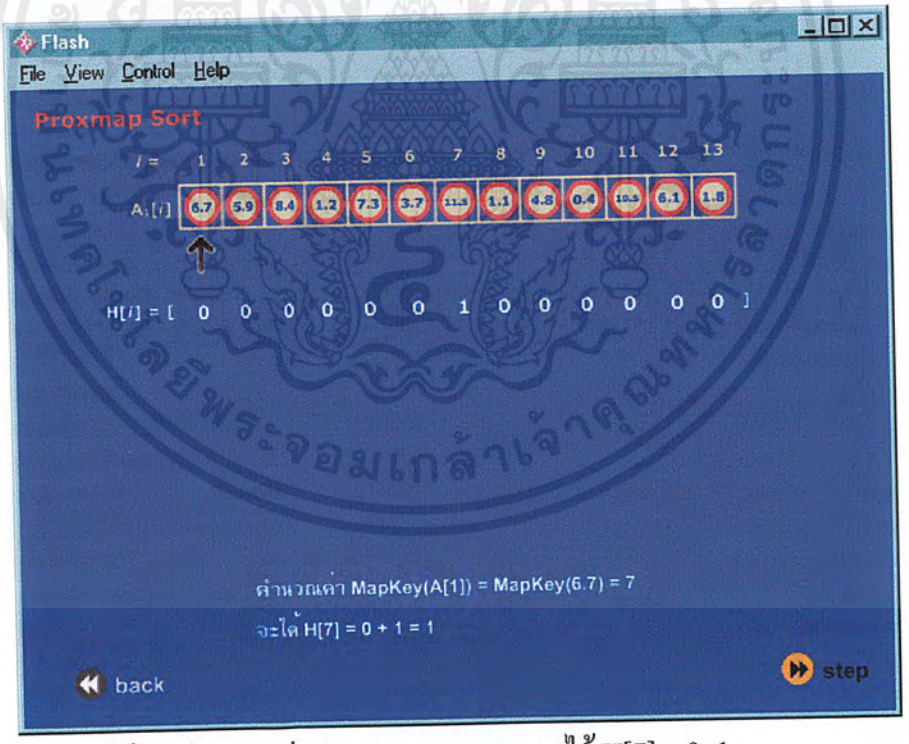


ภาพที่ 4.76 แสดงหน้าจอเริ่มต้นกับชุดข้อมูล A เพื่อทำการจัดเรียงด้วยวิธี Proxmap Sort

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

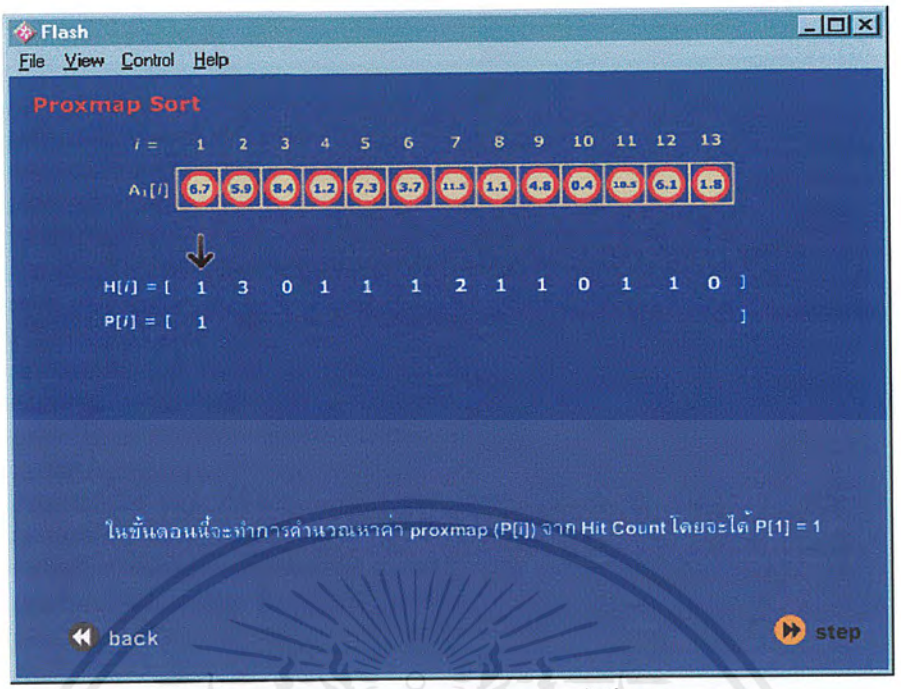


ภาพที่ 4.77 แสดงหน้าจอเริ่มต้นในการหาค่า Hit Counts จากสูตรที่กำหนดดังภาพ

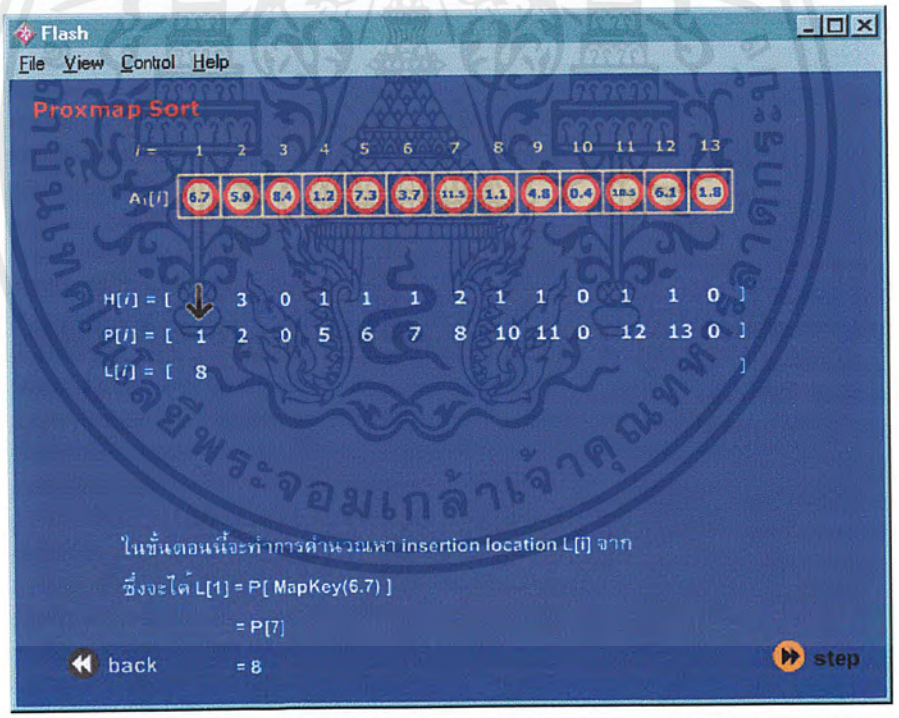


ภาพที่ 4.78 แสดงหน้าจอค่าการหาค่า Hit Countsจากสูตรจะได้ H[7] = 0+1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

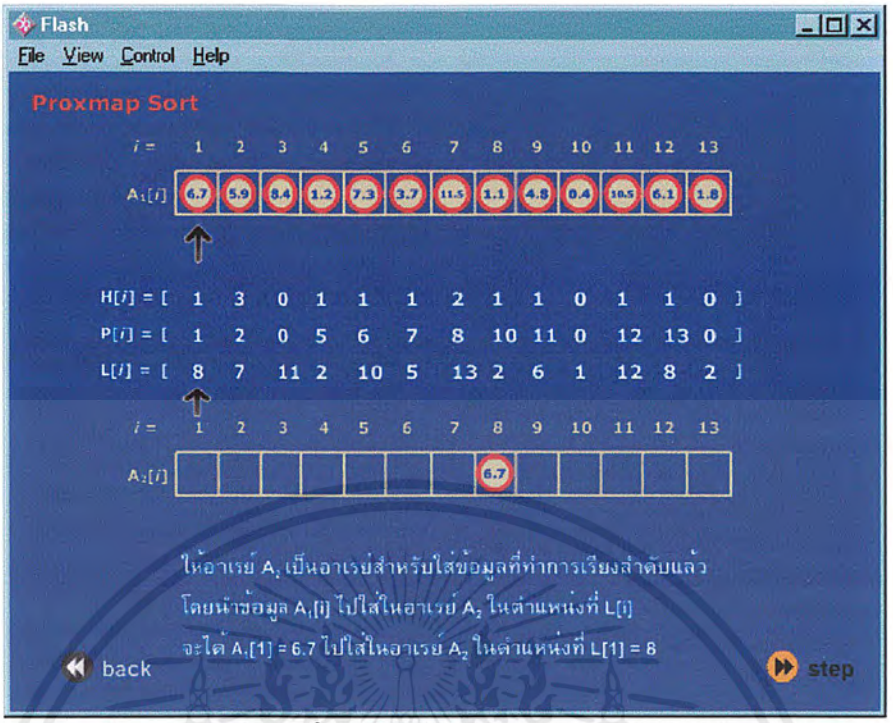


ภาพที่ 4.79 แสดงหน้าจอการหาค่า Proxmap (P[i]) จาก H[i]จะได้ P[1] = 1

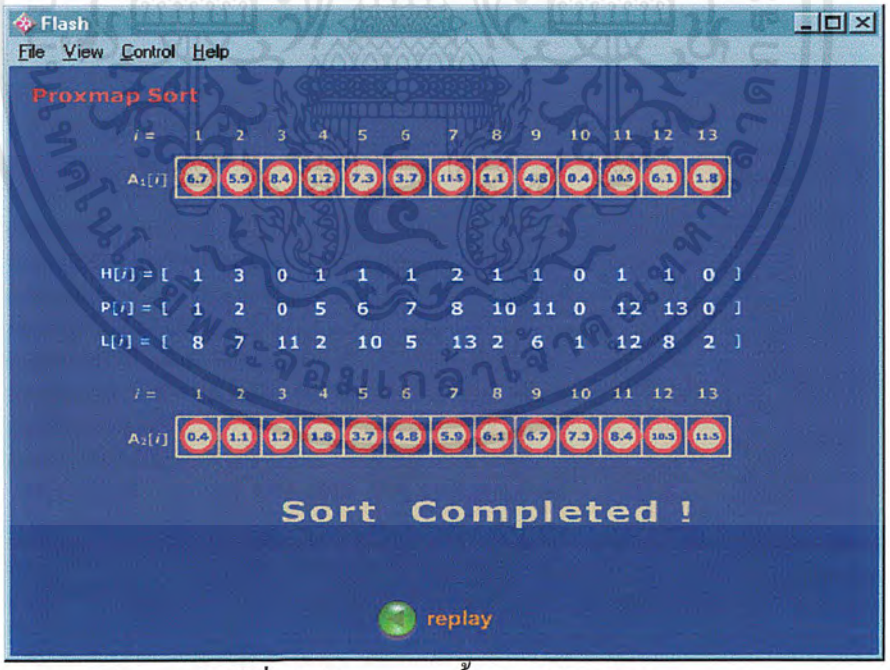


ภาพที่ 4.80 แสดงหน้าจอในการหาค่า Insertion Location จากสูตร L[i]=P[MapKey[i]]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.81 แสดงหน้าจอการใส่ข้อมูลที่เรียงลำดับแล้วโดยนำ $A_1[i]$ ไปใส่ไว้ใน A_2 ในตำแหน่ง L[i] เมื่อให้อาร์เรย์ A_2เป็นอาร์เรย์ผลลัพธ์

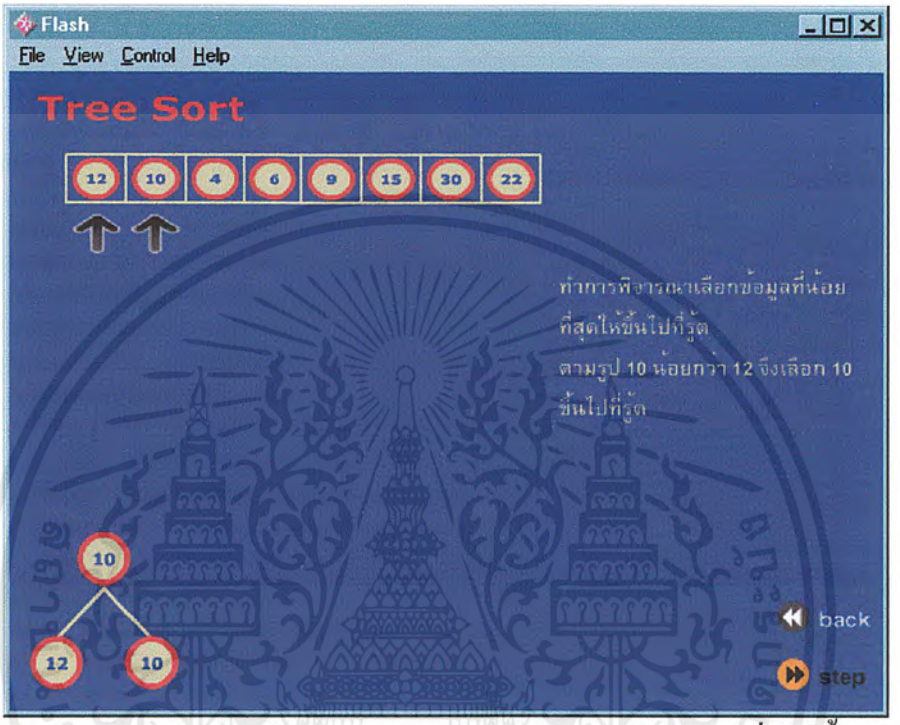


ภาพที่ 4.82 แสดงหน้าจอผลลัพธ์เมื่อดำเนินการตามขั้นตอนเดิมจนจบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

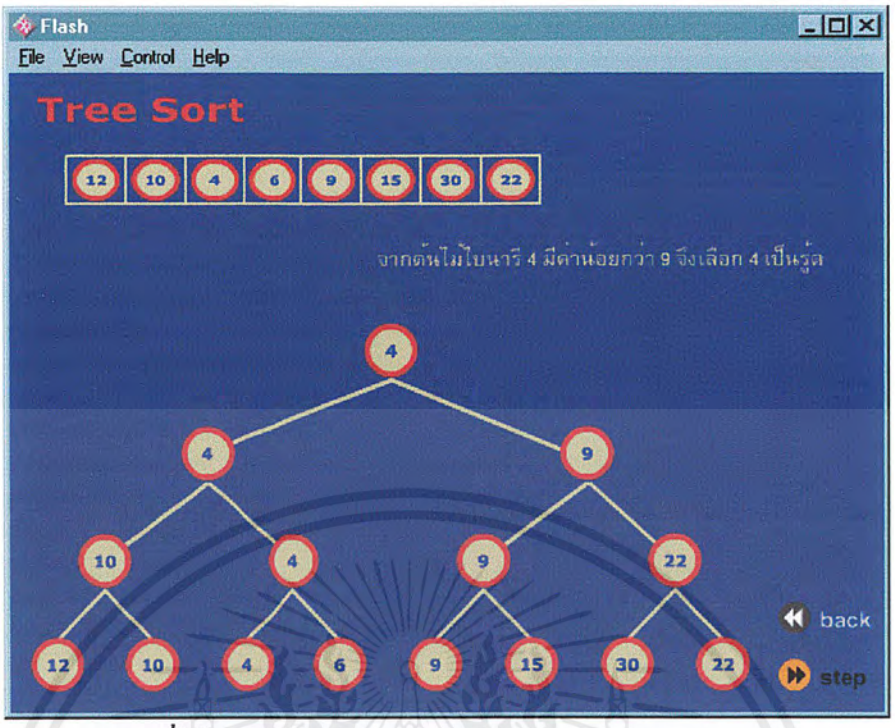
4.1.9 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Tree Sort

ในขั้นตอนแรกจะทำการสาธิตการสร้าง Binary Tree แบบที่มีค่าที่ Root Node เป็นค่าที่น้อยที่สุดจาก ชุดข้อมูล จากนั้น จึงนำค่าที่ Root Node ออกไปและทำการเลือกค่าที่น้อยที่สุดจากชุดข้อมูลที่เหลือมาไว้ที่ Root Node สามารถแสดงการทำงานได้ดังภาพต่อไปนี้

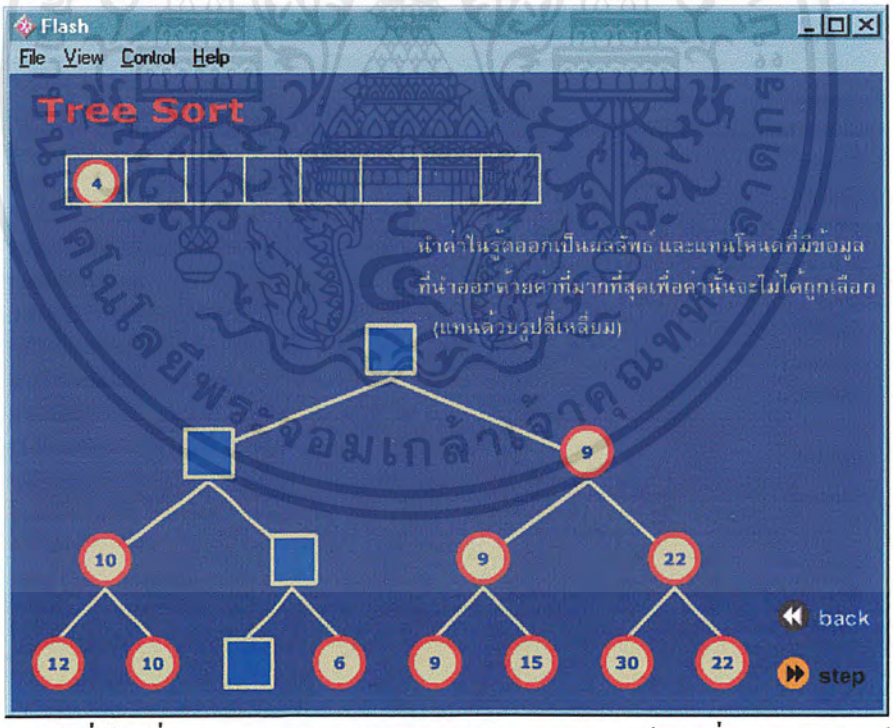


ภาพที่ 4.83 แสดงหน้าจอในการสร้าง Binary Tree โดยจับคู่ข้อมูลแล้วเลือกค่าที่น้อยขึ้นมาเป็น Root Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

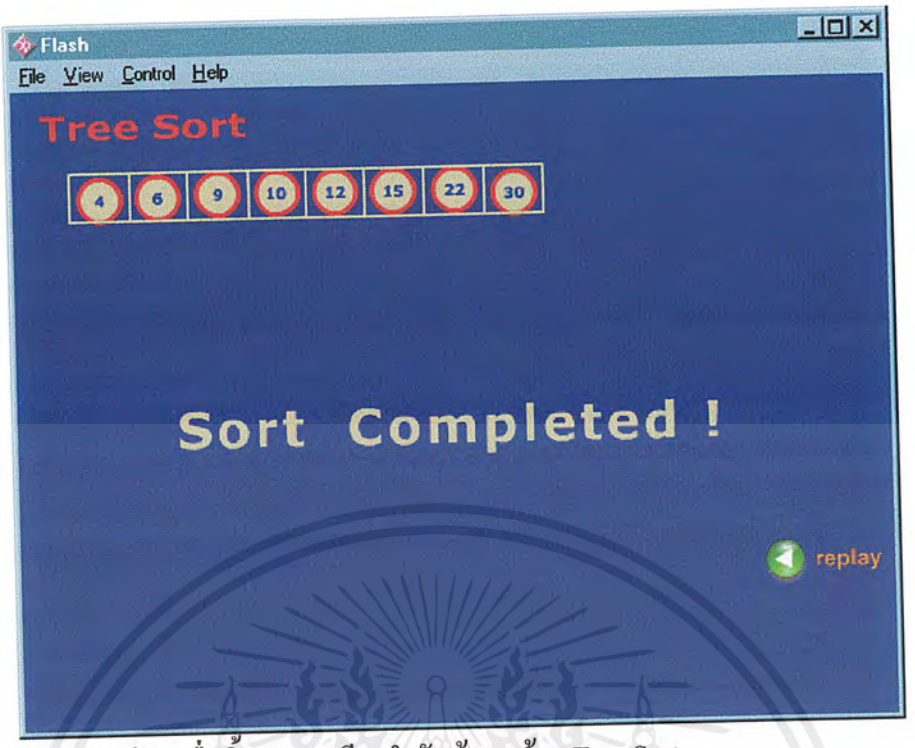


ภาพที่ 4.84 Binary Tree ที่ได้จากชุดข้อมูล 12 10 4 6 9 15 30 22



ภาพที่ 4.85 นำค่าที่น้อยที่สุดออกจาก Root Node และทำการแทนค่าข้อมูลที่ออกจาก Tree ไปนั้นด้วยตัวเลขที่มีค่ามากๆในที่นี้แทนด้วยสัญลักษณ์ สี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.86 แสดงหน้าจอเมื่อสิ้นสุดการเรียงลำดับข้อมูลด้วย Tree Sort

4.1.10 โปรแกรมสาธิตการจัดเรียงข้อมูลด้วย Quick Sort

จะมีความสัมพันธ์ คือ $\{ S_1 \} < X < \{ S_2 \}$ เมื่อ $\{ S_1 \}$ คือเซ็ตของข้อมูลที่น้อยกว่า ตัวแบ่ง X_1

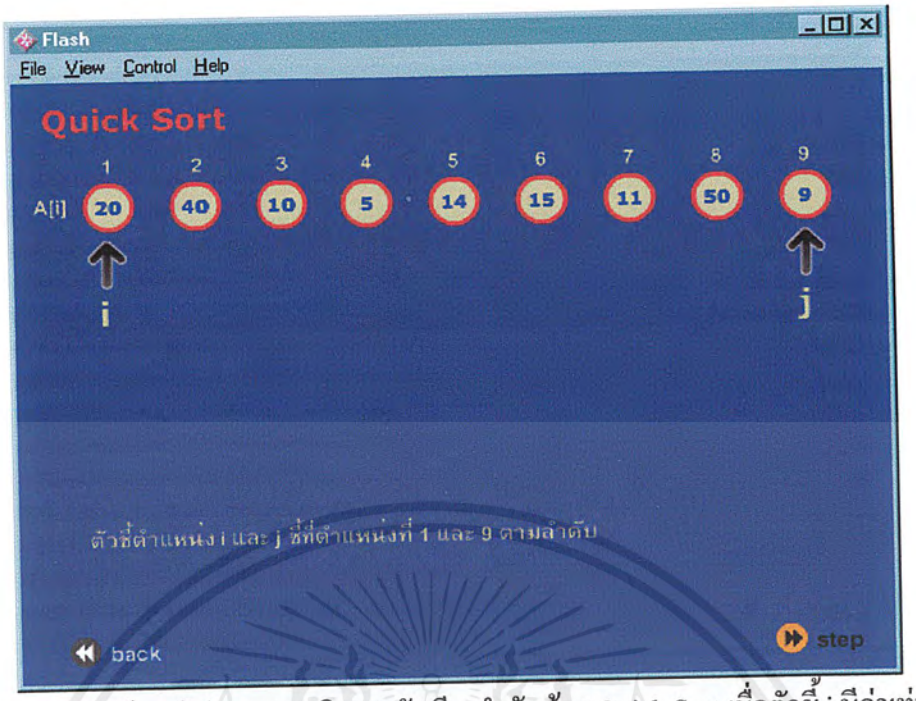
$\{ S_2 \}$ คือเซ็ตของข้อมูลที่มากกว่า ตัวแบ่ง X_1

มีขั้นตอนการทำงาน

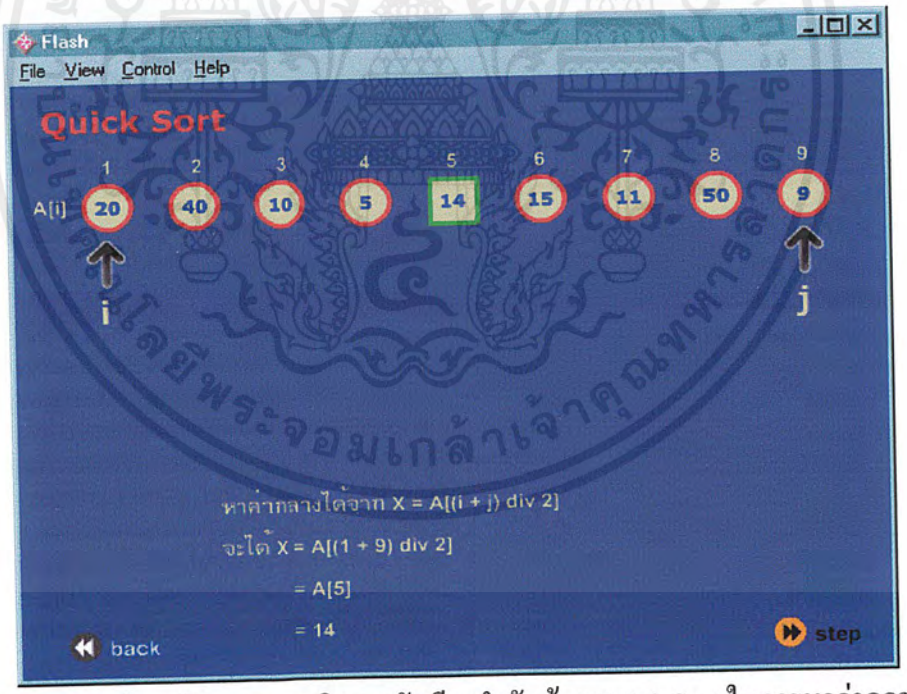
1. เลือกข้อมูลตัวใดตัวหนึ่งเป็นตัวแบ่ง เช่น ใช้ข้อมูลตัวแรกคือ A[1]
2. นำข้อมูลที่มีค่าน้อยกว่าตัวแบ่งมาวางไว้ทางซ้ายของตัวแบ่ง
3. นำข้อมูลที่มีค่ามากกว่าตัวแบ่งมาวางไว้ทางขวาของตัวแบ่ง
4. ทำซ้ำในข้อ 1 ถึง ข้อ 3 กับกลุ่มข้อมูลที่อยู่ทางซ้ายและทางขวาของตัวแบ่ง ในลักษณะเดียวกันจน แบ่งกลุ่มข้อมูลไม่ได้อีกแล้ว
5. นำข้อมูลทั้งหมดทางด้านซ้าย ด้านขวา และตัวแบ่ง มารวมกันเป็นผลลัพธ์

สามารถแสดงการทำงานการจัดเรียงข้อมูลด้วยวิธี Quick Sort ได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



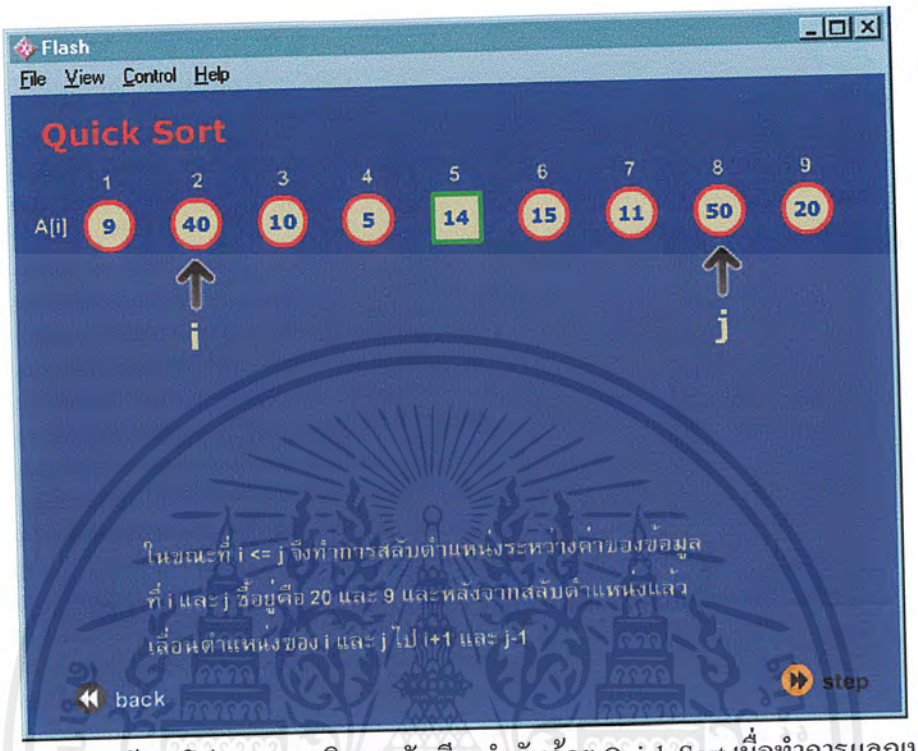
ภาพที่ 4.87 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงลำดับด้วย Quick Sort เมื่อตัวชี้ i มีค่าเท่ากับ 1 ชี้ที่ A[1] และตัวชี้ j=9 ชี้ที่ A[9]



ภาพที่ 4.88 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงลำดับด้วย Quick Sort ในการหาค่ากลางโดยกำหนดให้ค่ากลางคือ X=A[(i+j)div2] ซึ่งจากชุดข้อมูลนี้ค่ากลางมีค่าเท่ากับ 14 ทำการเปรียบเทียบค่า A[i] กับ ค่ากลางเนื่องจาก A[i] > X ดังนั้นจึงพิจารณา A[j] < X ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแลกเปลี่ยนค่า ณ. ตำแหน่ง A[i] และ A[j] พร้อมทั้งเพิ่มค่า i และลดค่า j ดังแสดงใน ภาพที่ 4.89

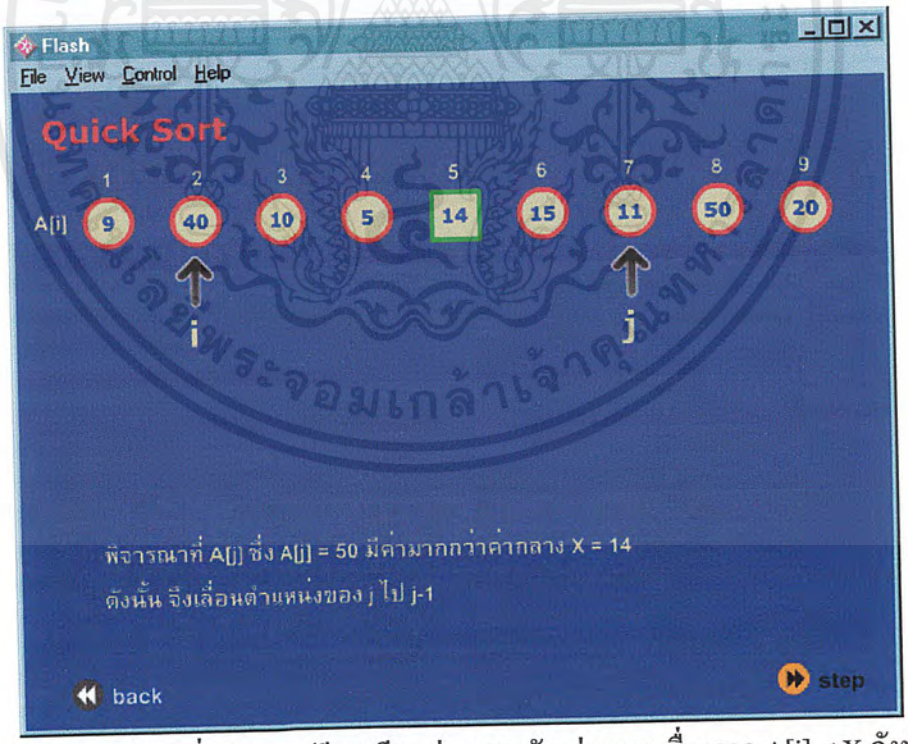


ภาพที่ 4.89 แสดงหน้าจอโปรแกรมสาธิตการจัดเรียงลำดับด้วย Quick Sort เมื่อทำการแลกเปลี่ยนค่า ณ. ตำแหน่ง A[i] และ A[j]

จากภาพที่ 4.89 ทำการเปรียบเทียบค่า A[i] กับ ค่ากลางเนื่องจาก A[i] > X ดังนั้นมาพิจารณาค่าที่ A[j] แต่เนื่องจากค่าที่ A[j] มีค่ามากกว่า X ดังนั้นจึงลดค่า j ลงดังแสดงในภาพที่ 4.90

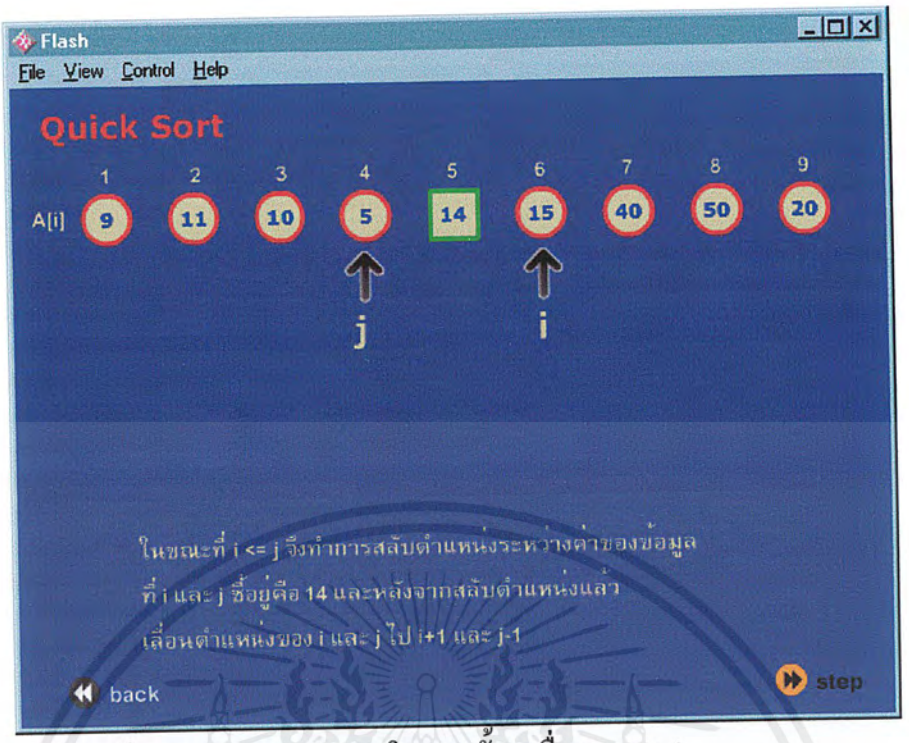
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.90 แสดงหน้าจอเมื่อ A[i]>X ดังนั้นมาพิจารณาค่าที่ A [j] แต่เนื่องจากค่าที่A[j] > X ดังนั้นจึงลดค่า j



ภาพที่ 4.91 แสดงหน้าจอเมื่อทำการเปรียบเทียบค่า A[j] กับ ค่ากลางเนื่องจาก A[j] < X ดังนั้น ทำการแลกเปลี่ยนค่าระหว่างตำแหน่ง A[i] กับ A[j]

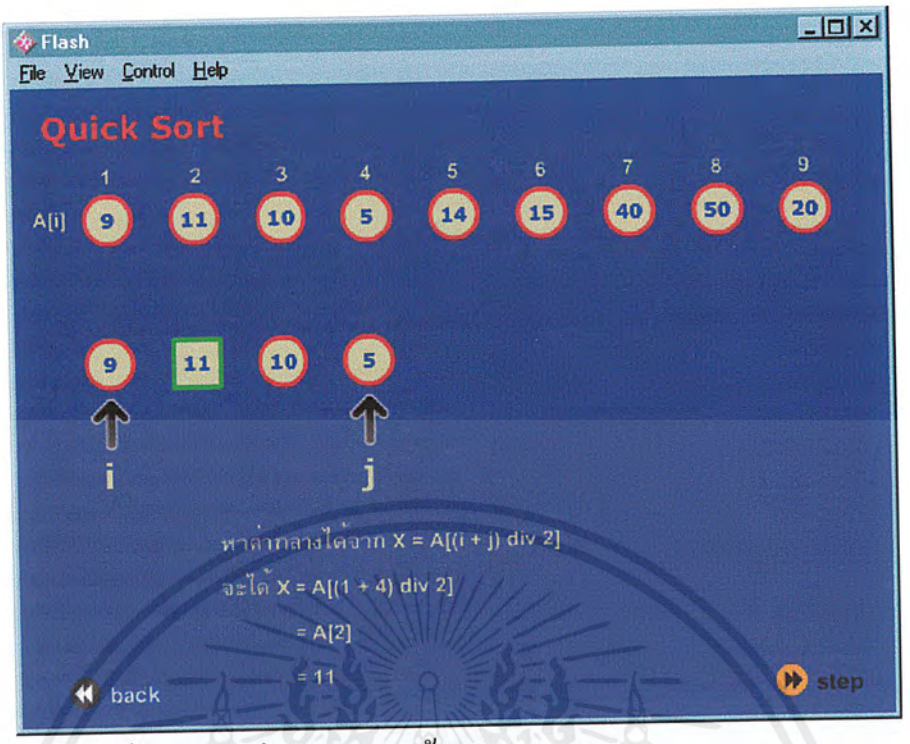
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



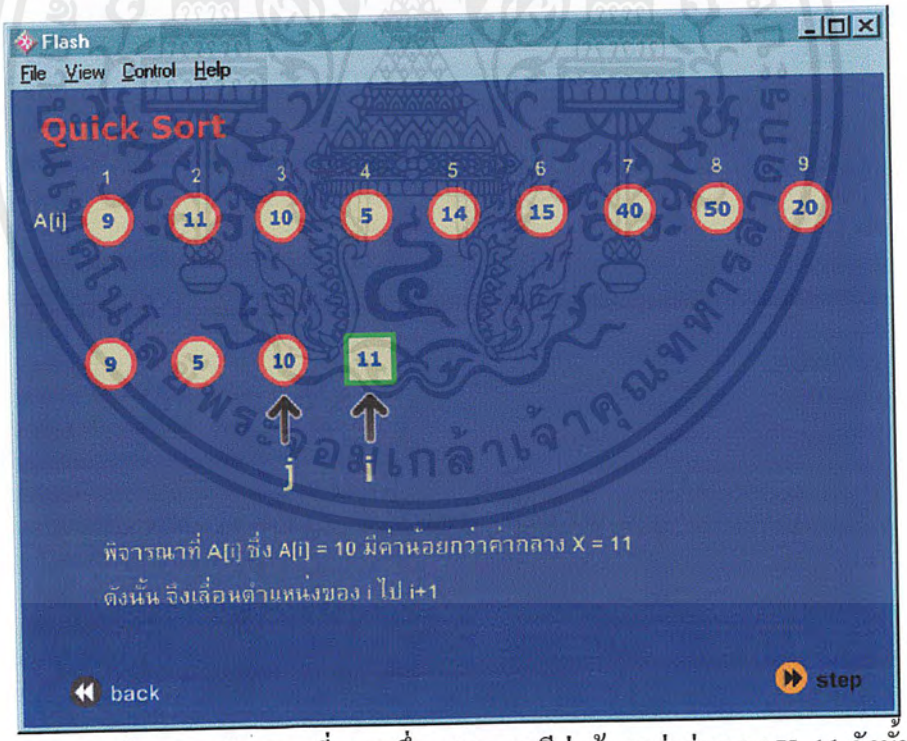
ภาพที่ 4.92 แสดงหน้าจอการหยุดการทำงานในรอบนั้นๆเนื่องจาก j > i

ภาพที่ 4.93 แสดงหน้าจอในดำเนินการ Quick Sort กับข้อมูลในชุดด้านซ้ายคือ 9 11 10 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

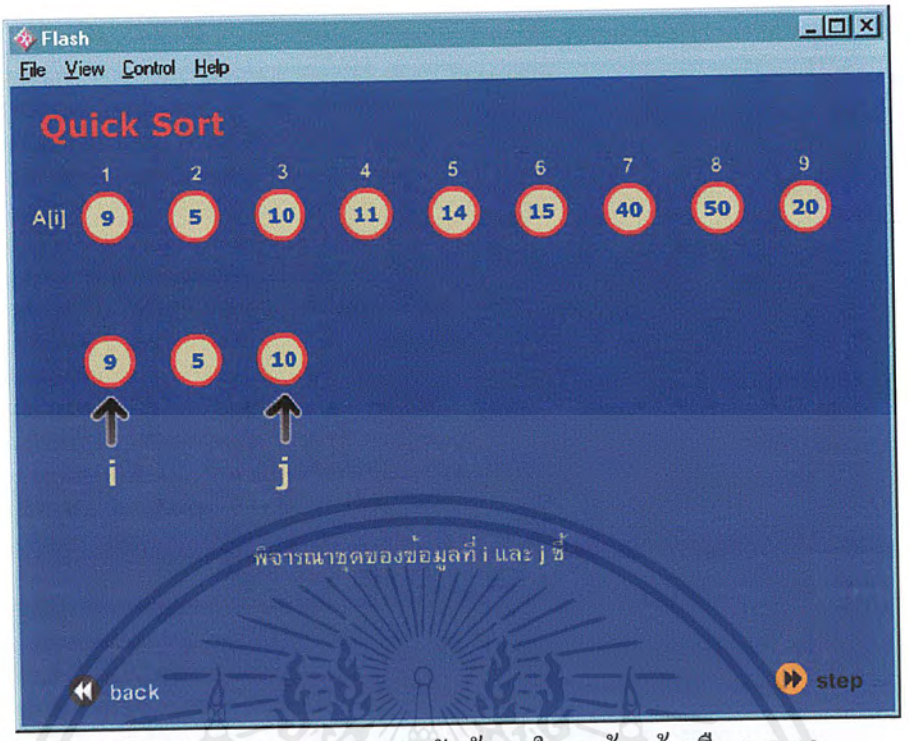


ภาพที่ 4.94 แสดงหน้าจอการหาค่ากลางของชุดข้อมูล 9 11 10 5

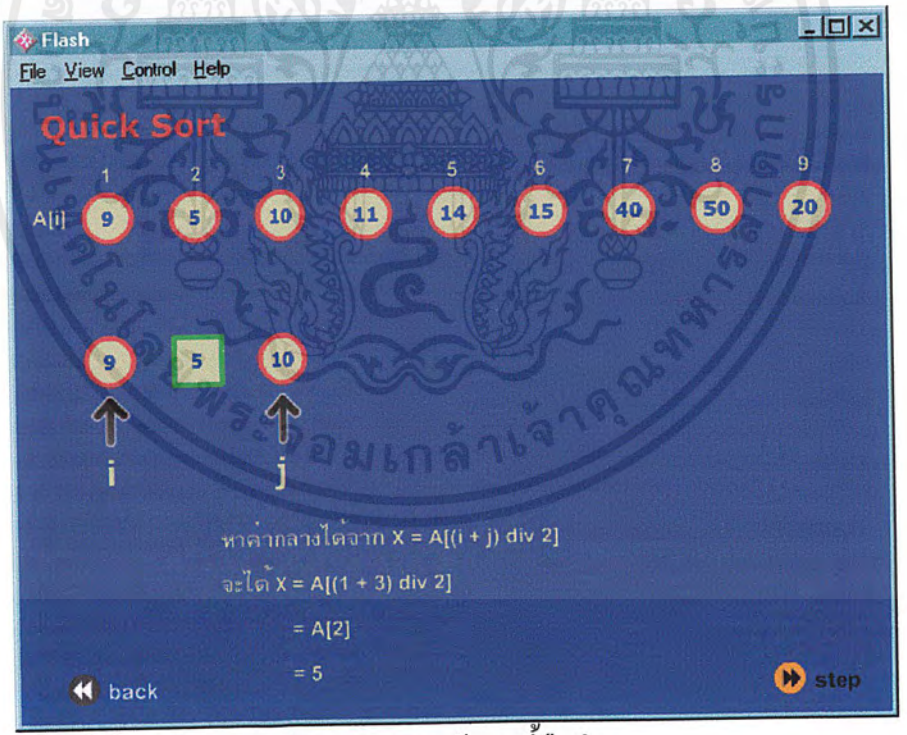


ภาพที่ 4.95 แสดงหน้าจอการพิจารณาที่ A[i] ซึ่ง A[i] =10 มีค่าน้อยกว่าค่ากลาง X=11 ดังนั้นจึงเลื่อนตำแหน่งของ i ไปที่ i +1 เมื่อ j > i หยุดการพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

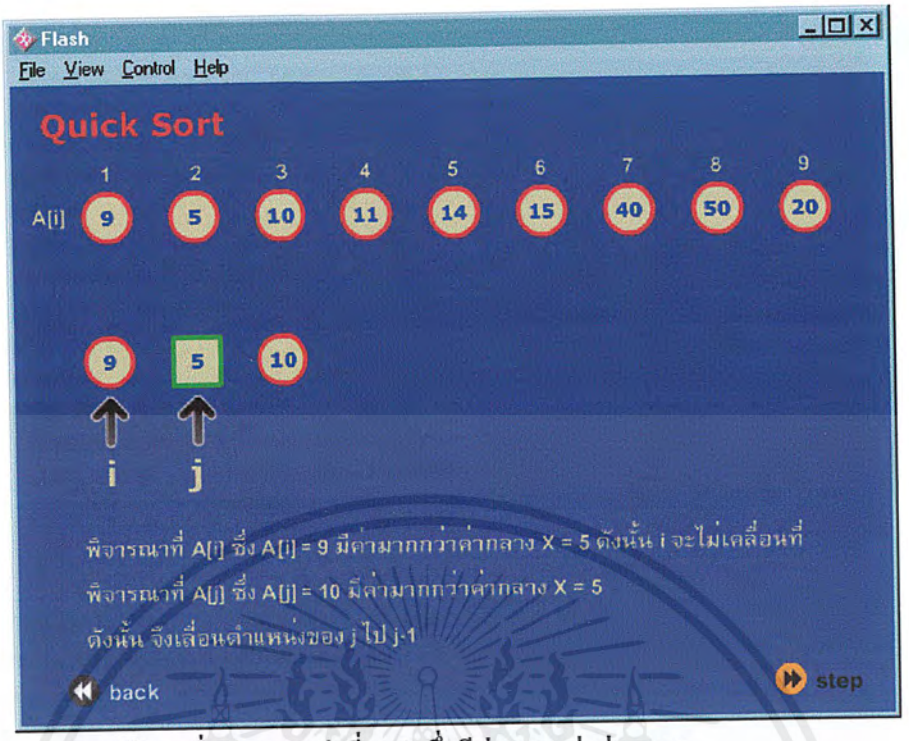


ภาพที่ 4.96 แสดงหน้าจอดำเนินการ Quick Sort กับข้อมูลในชุดด้านซ้ายคือ 9 5 10

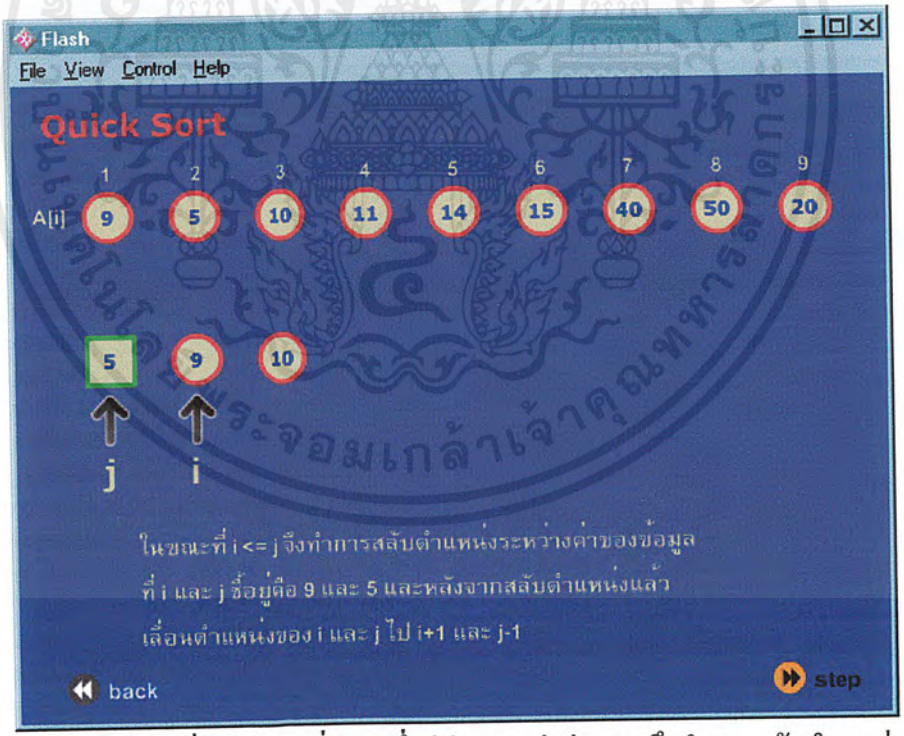


ภาพที่ 4.97 แสดงหน้าจอในการหาค่ากลางของชุดข้อมูลนี้คือค่า 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

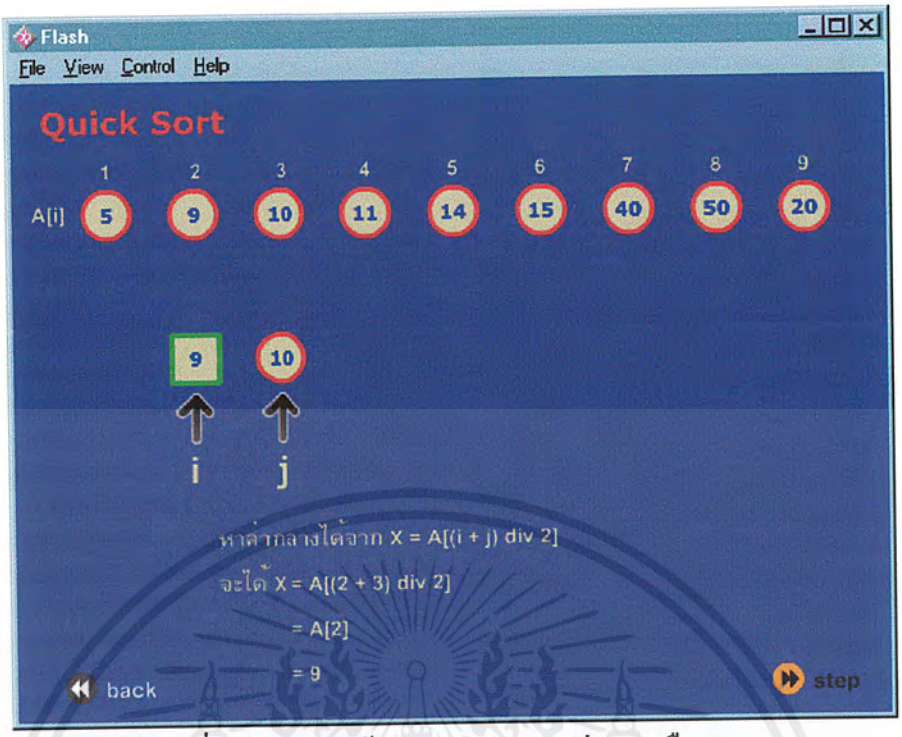


ภาพที่ 4.98 แสดงหน้าจอเมื่อพิจารณาค่าที่ A[i] ซึ่งมีค่ามากกว่าค่ากลาง

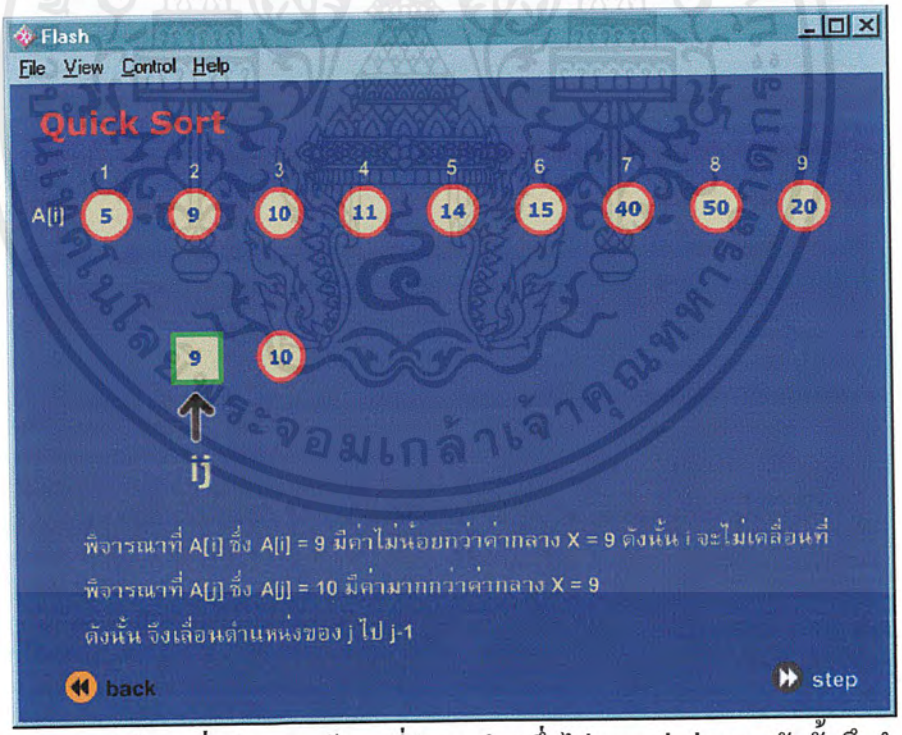


ภาพที่ 4.99 แสดงหน้าจอเมื่อพิจารณาที่ A[i] ซึ่งมีค่ามากกว่าค่ากลางจึงทำการสลับตำแหน่งและเมื่อสลับตำแหน่งแล้วเพิ่มค่า i และลดค่า j และเนื่องจาก i มากกว่า j จึงหยุดการพิจารณา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

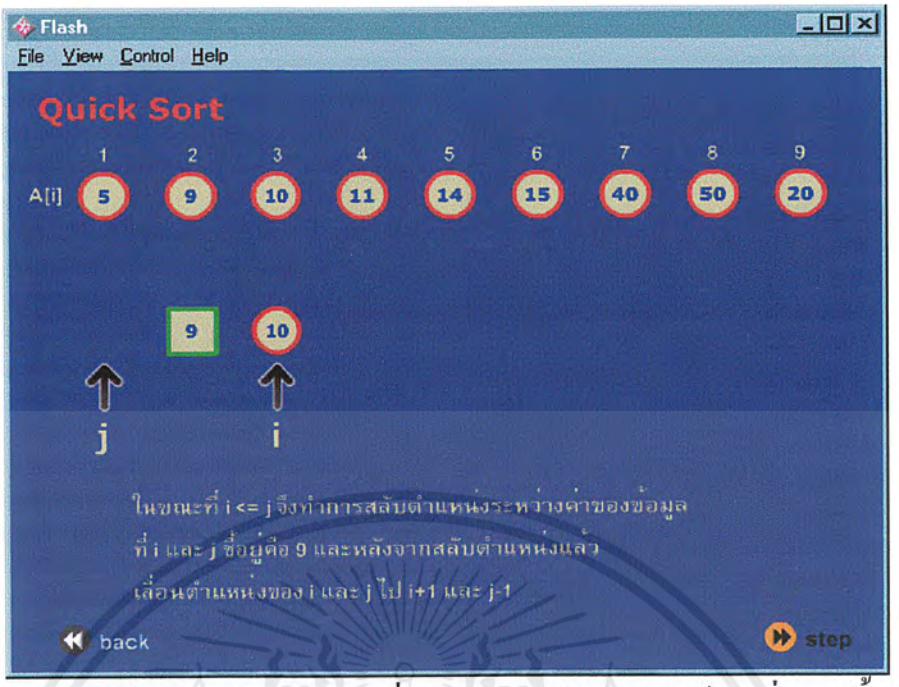


ภาพที่ 4.100 แสดงหน้าจอเมื่อพิจารณาชุดข้อมูล 9 10 และหาค่ากลางคือ 9

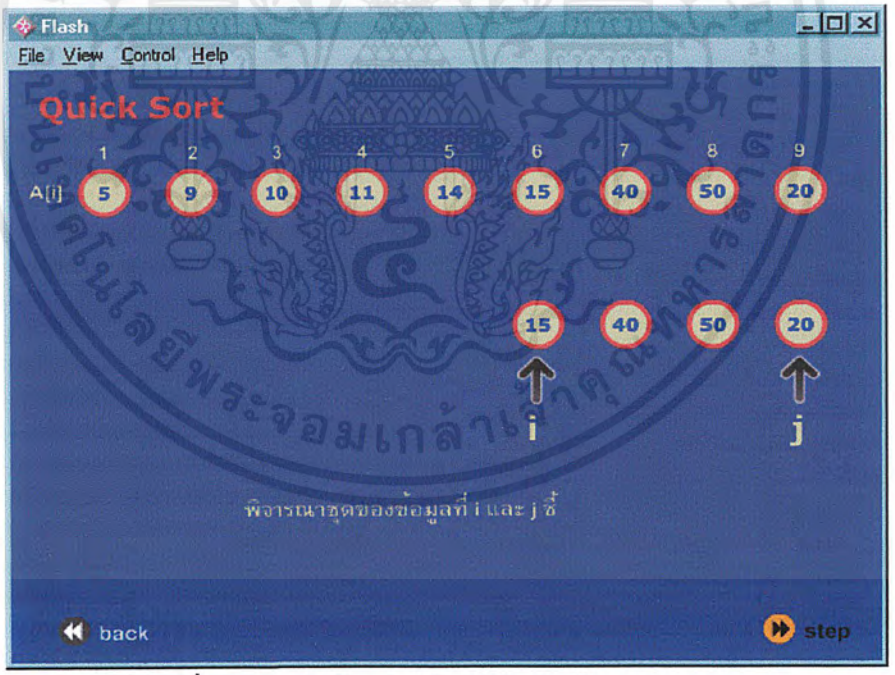


ภาพที่ 4.101 แสดงหน้าจอเมื่อพิจารณาข้อมูลที่ตำแหน่ง i ซึ่งไม่มากกว่าค่ากลางดังนั้นจึงทำการพิจารณาค่าที่ตำแหน่ง j และเนื่องจาก A[j] มีค่ามากกว่าค่ากลางจึงทำการลดค่า j

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.102 แสดงหน้าจอการทำงานในขณะที่ i<=j ทำการสลับตำแหน่งข้อมูลที่และj ชี้อยู่คือ 9 และทำการเพิ่มค่าi และลดค่า j



ภาพที่ 4.103 แสดงหน้าจอเมื่อทำการดำเนินการด้วยวิธีเดิมกับข้อมูล 15 40 50 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.104 แสดงหน้าจอการทำงานเมื่อดำเนินการจนได้ชุดข้อมูลที่เรียงลำดับแล้ว

4.2 หลักการทำงานของโปรแกรมจำลองการจัดเรียงข้อมูล

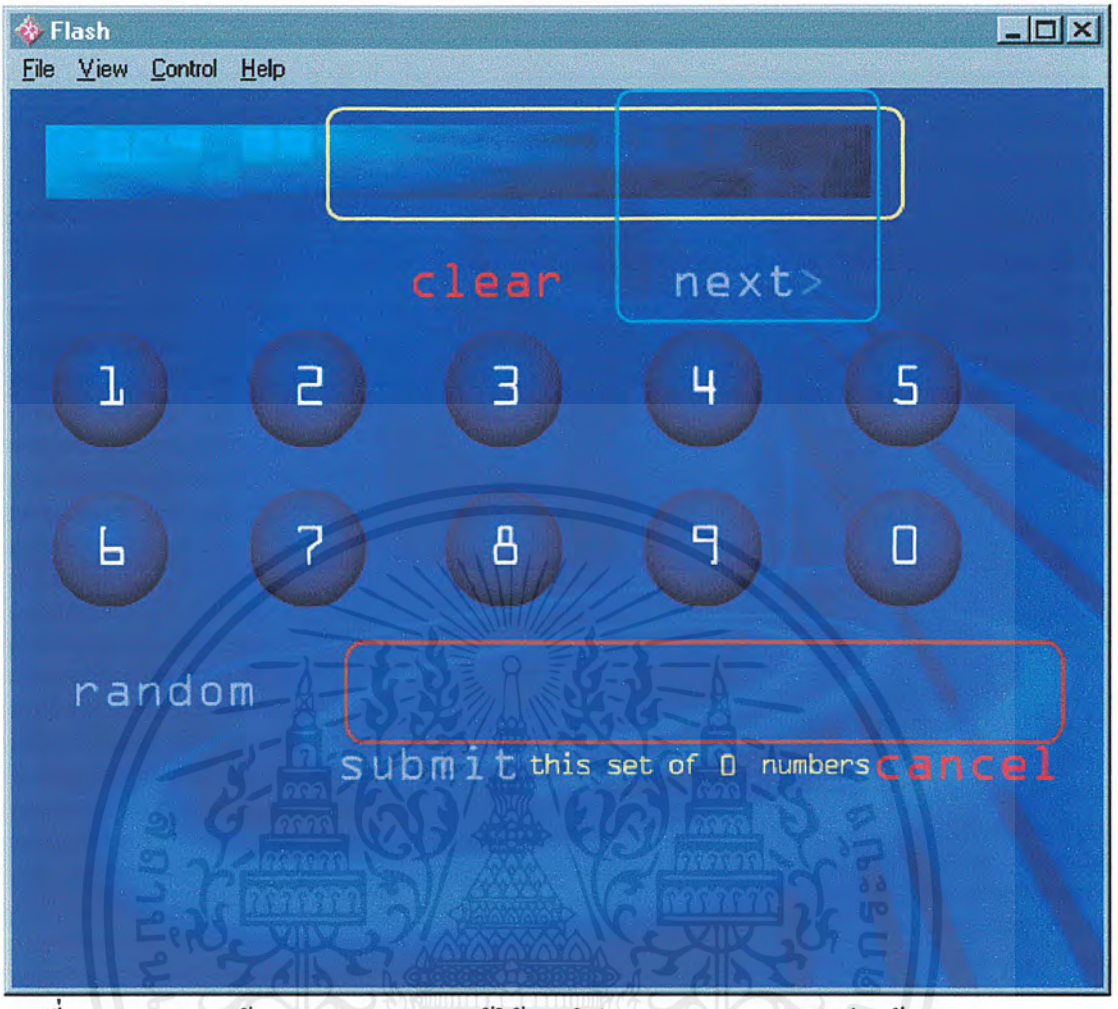
4.2.1 ส่วนประกอบของหน้าจอการทำงานของโปรแกรมจำลองการจัดเรียงข้อมูล

โปรแกรมการจัดเรียงข้อมูลมีวัตถุประสงค์เพื่อให้ผู้ใช้สามารถเห็นภาพการใช้งานโปรแกรมการจัดเรียงข้อมูลในแต่ละวิธีการจัดเรียงข้อมูลโดยผู้ใช้สามารถป้อนชุดของข้อมูลผ่านหน้าจอ Interface ซึ่งมีความสวยงามทันสมัยและสะดวกต่อการใช้งานดังภาพที่ 4.104 ซึ่งแสดงหน้าจอรับชุดของข้อมูลซึ่งสามารถรับข้อมูลได้มากสุดจำนวน 10 ค่าต่อหนึ่งชุดของข้อมูลซึ่งตัวเลขแต่ละค่าจะต้องมีไม่เกิน 4 หลักโดยผู้ใช้สามารถป้อนข้อมูลแต่ละชุดโดยการ Click ที่ปุ่มตัวเลข จากนั้น กดปุ่ม Next เพื่อใส่ค่าต่อไป ซึ่งค่าที่ได้ใส่ไปแล้วจะปรากฏอยู่ในกรอบแสดงด้านล่าง พร้อมจำนวนค่าของข้อมูลในชุดข้อมูลนั้นๆ

เมื่อผู้ใช้ป้อนชุดข้อมูลครบทั้ง 10 ตัวแล้ว ผู้ใช้จะไม่สามารถเพิ่มเติมเลขใดๆได้อีก และจำนวนข้อมูลในชุดข้อมูลต้องมีมากกว่า 1 ค่ามิเช่นนั้นจะปรากฏผลดังหน้าจอ 4.105

เมื่อผู้ใช้ป้อนข้อมูลครบดังต้องการให้กดปุ่ม Submit เพื่อแสดงขั้นตอนการเรียงลำดับชุดข้อมูลนั้นๆตามแต่ละวิธีการจัดเรียงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



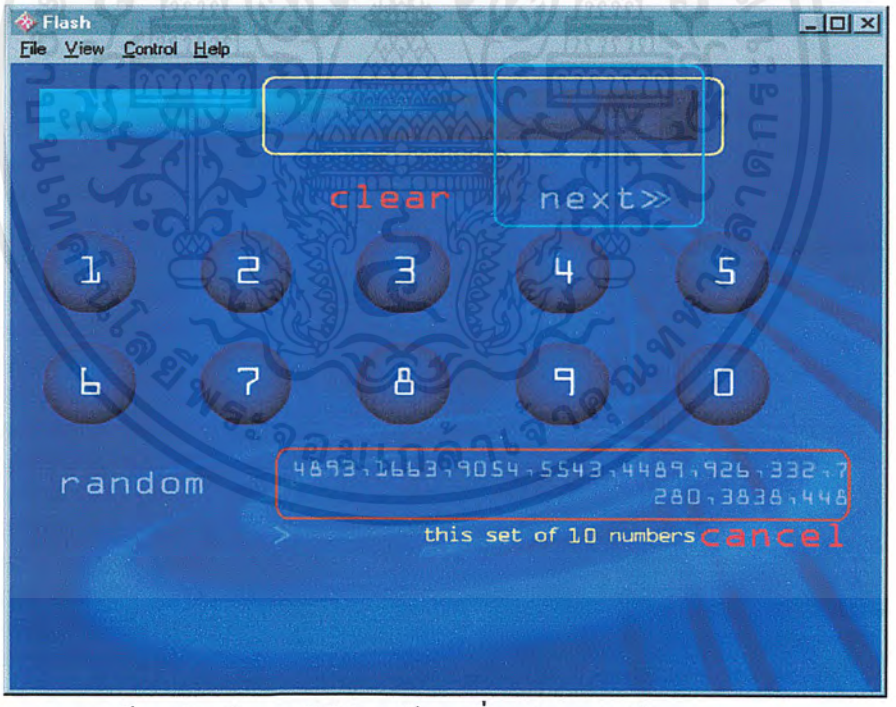
ภาพที่ 4.105 แสดงหน้าจอรับชุดตัวเลขจากผู้ใช้ของโปรแกรมจำลองการจัดเรียงข้อมูล

ส่วนประกอบของหน้าจอรับชุดข้อมูล

1. ปุ่มตัวเลข 0-9 ผู้ใช้สามารถ Click ที่ปุ่มตัวเลขแต่ละปุ่มเพื่อทำการ Input ชุดของข้อมูล
2. ปุ่ม Next เมื่อผู้ใช้ใส่ข้อมูลเสร็จหนึ่งค่า และต้องการใส่ค่าต่อไปให้กดปุ่มนี้
3. ปุ่ม Clear เมื่อผู้ใช้ต้องการยกเลิกการกรอกข้อมูลค่าที่ Input ตัวล่าสุดและยังไม่ได้กด Next ให้ผู้ใช้กดปุ่ม Clear เพื่อลบค่านั้นๆทิ้ง
4. ปุ่ม Cancel เมื่อผู้ใช้ต้องการลบชุดข้อมูลทั้งหมดที่ได้ Input ไปแล้วทุกค่า
5. ปุ่ม Submit เมื่อผู้ใช้ตกลงที่จะทำการจัดเรียงข้อมูลชุดที่ได้ทำการ Input เข้าไป
6. ตัวเลขแสดงจำนวนข้อมูลทั้งหมดในชุดข้อมูลนั้นๆ
7. ช่องแสดงค่าแต่ละค่าในชุดข้อมูลนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.106 แสดงหน้าจอเมื่อชุดตัวเลขจากผู้ใช้มีจำนวนข้อมูลเพียง 1 ตัวซึ่งไม่สามารถทำการจัดเรียง

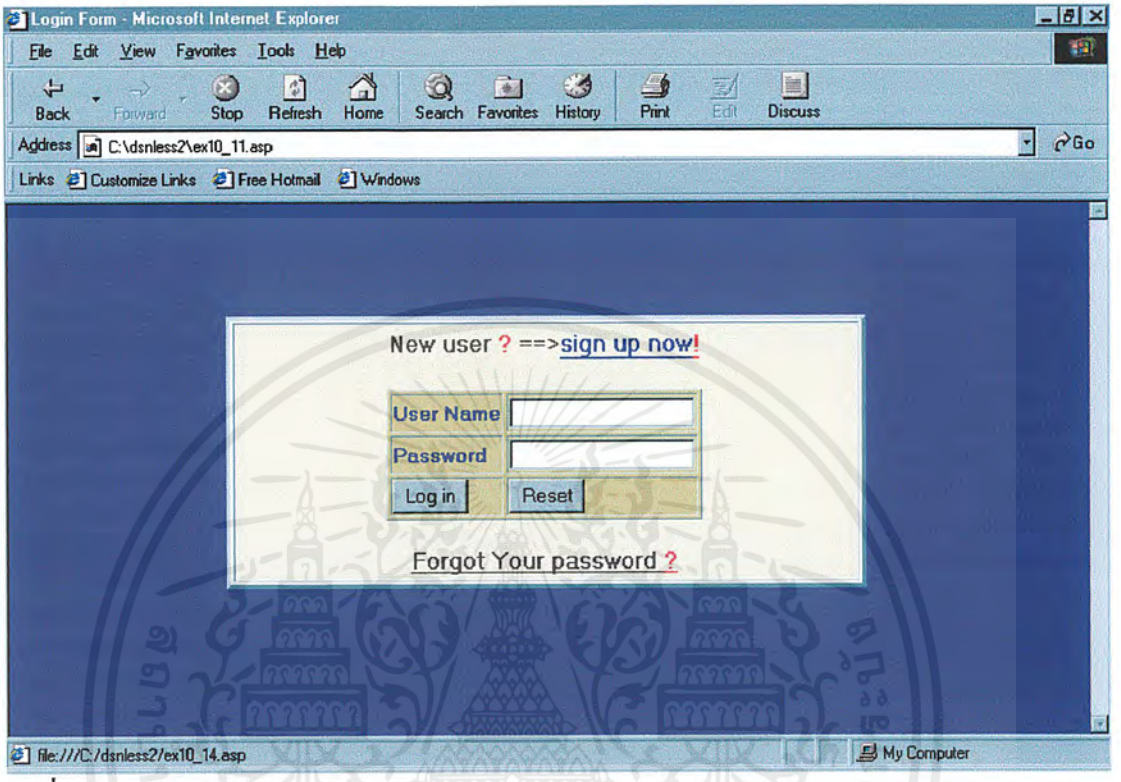


ภาพที่ 4.107 แสดงหน้าจอตัวอย่างการใส่ค่าชุดข้อมูลที่จะทำการจัดเรียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทำงานของระบบแบบฝึกหัด

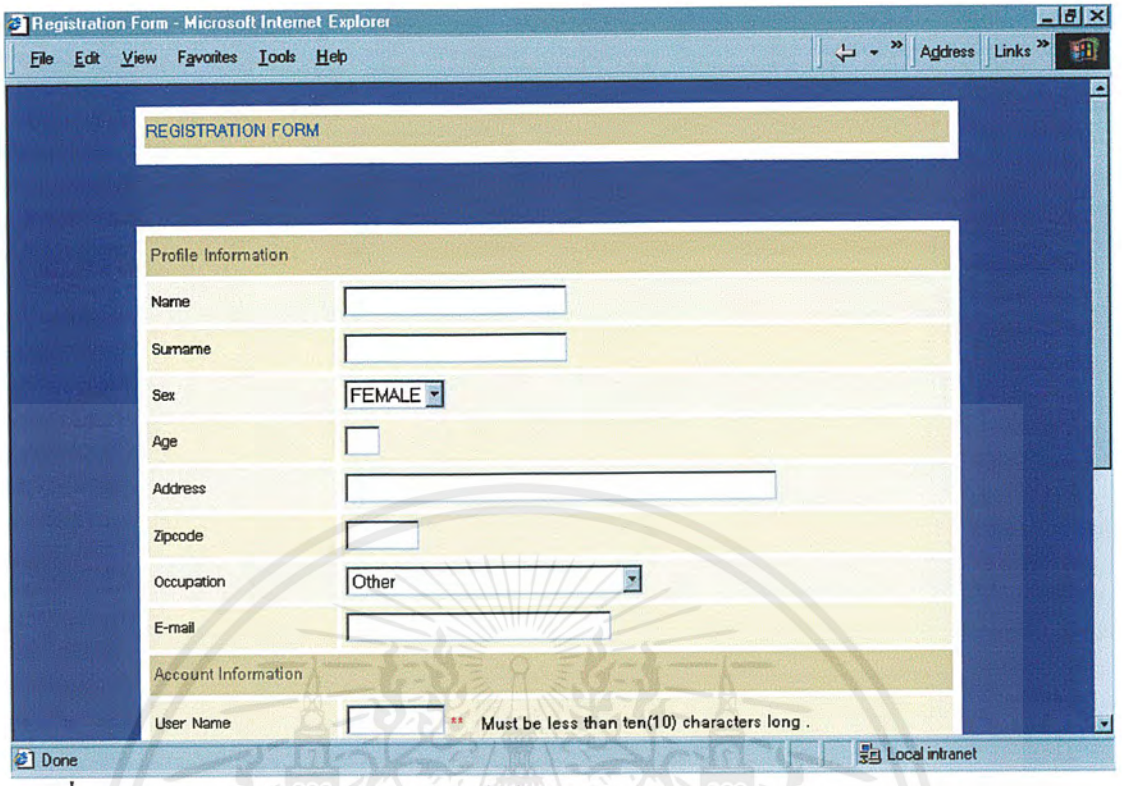
4.5.3 แสดงผลการทำงานของหน้าจอหลักของระบบแบบฝึกหัด



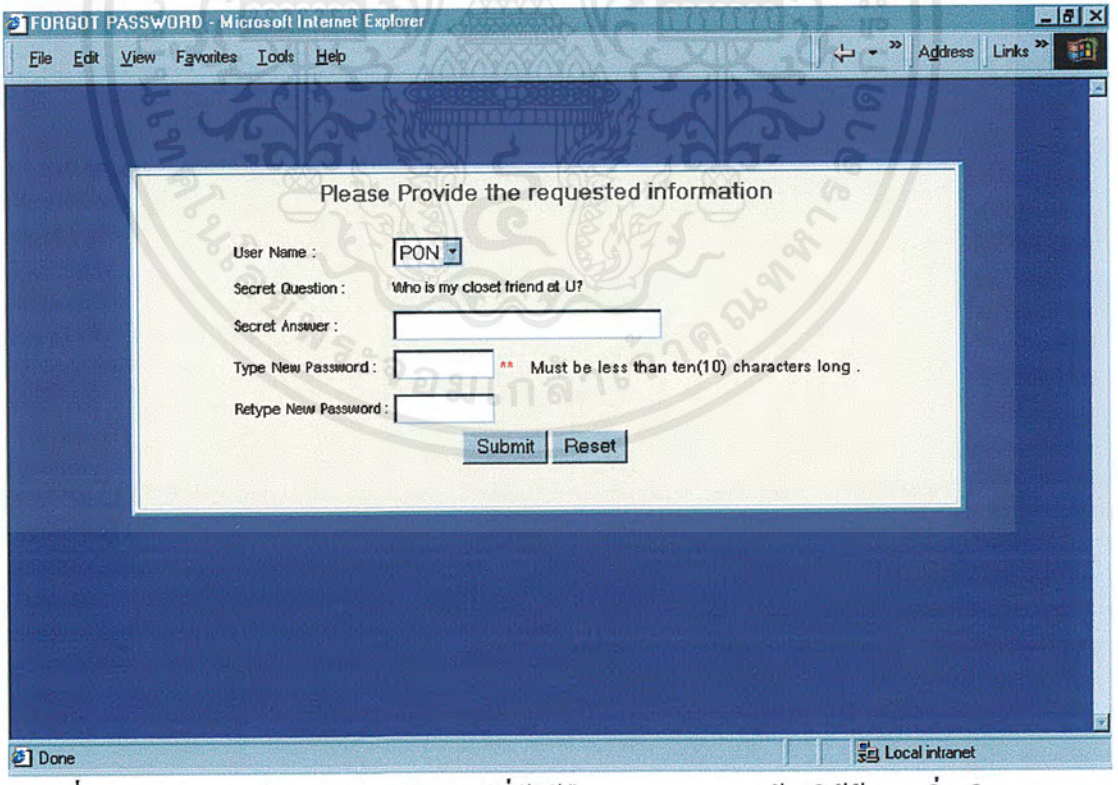
ภาพที่ 4.108 แสดงหน้าจอการ Login User

หน้าจอ Log in User นี้ผู้ใช้งานโปรแกรมจะต้องทำการใส่ User Name และ Password ของตน ซึ่งระบบจะทำการตรวจเช็คกับฐานข้อมูลถ้าถูกต้อง ผู้ใช้ก็จะสามารถผ่านเข้าใช้งานระบบได้ และในกรณีที่เป็นผู้ใช้ใหม่ไม่มี User Name และ Password ผู้ใช้จะต้องทำการลงทะเบียนโดย การ Click ที่ข้อความ Sign Up Now ซึ่งจะปรากฏหน้าจอดังภาพที่ 4.108 และหากผู้ใช้ที่มี User Name และมี Password แล้วแต่ลืม Password ของตนก็สามารถเข้าสามารถเข้าใช้ระบบได้โดยการ Click ที่ Forgot Your Password จะปรากฏหน้าจอดังภาพที่ 4.109 ซึ่งผู้ใช้จะต้องให้ข้อมูลต่างๆได้แก่ User Name และสามารถตอบคำถามจากโจทย์ที่ผู้ใช้ได้ตั้งเอาไว้ในครั้งที่ทำการลงทะเบียนได้ถูกต้อง ซึ่งระบบจะอนุญาตให้ผู้ใช้ที่ตอบคำถามได้ถูกต้องนี้สามารถเปลี่ยน Password ได้โดยใช้ User Name เดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

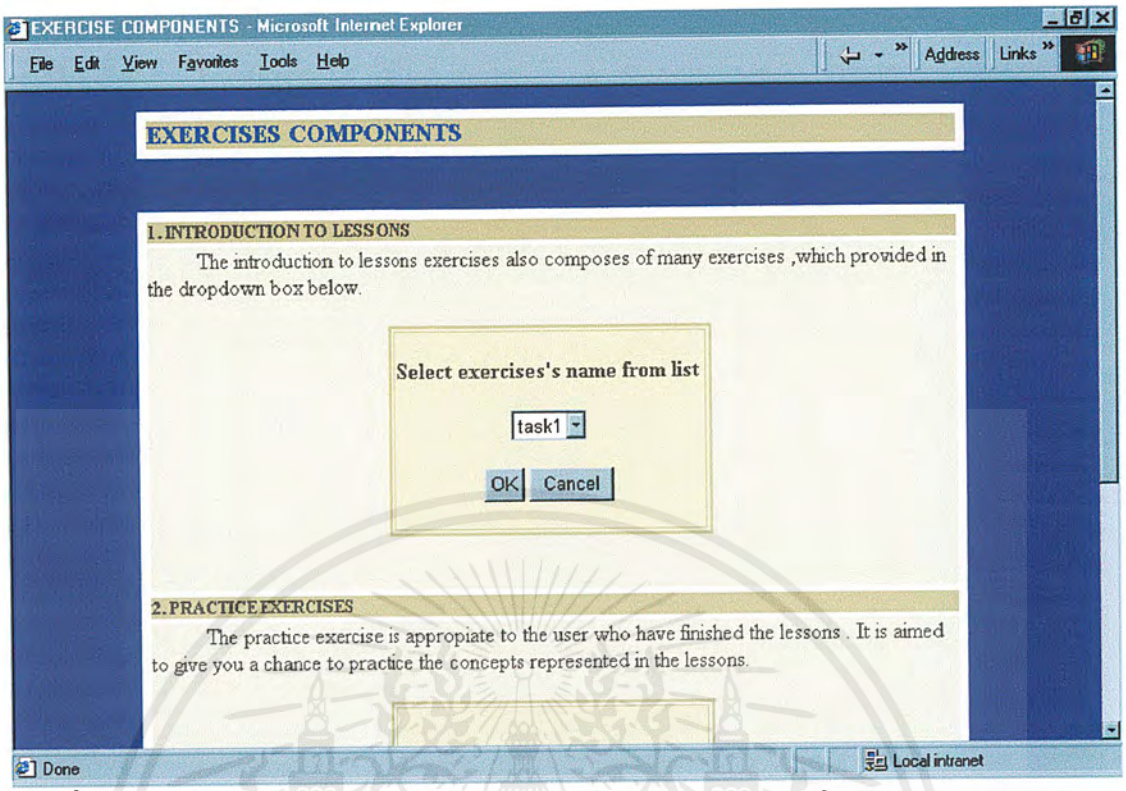


ภาพที่ 4.109 แสดงหน้าจอการลงทะเบียนผู้ใช้โปรแกรม

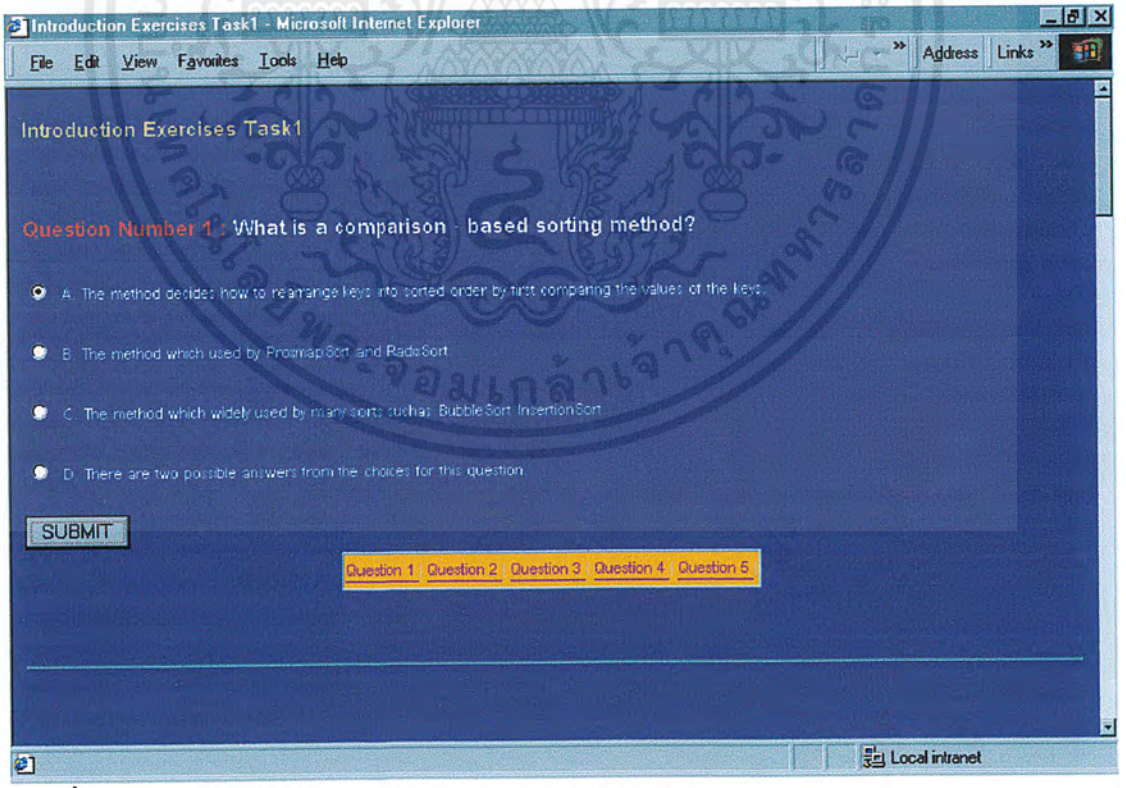


ภาพที่ 4.110 แสดงหน้าจอแสดงผลในกรณีที่ผู้ใช้ลืม Password และต้องให้ข้อมูลเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

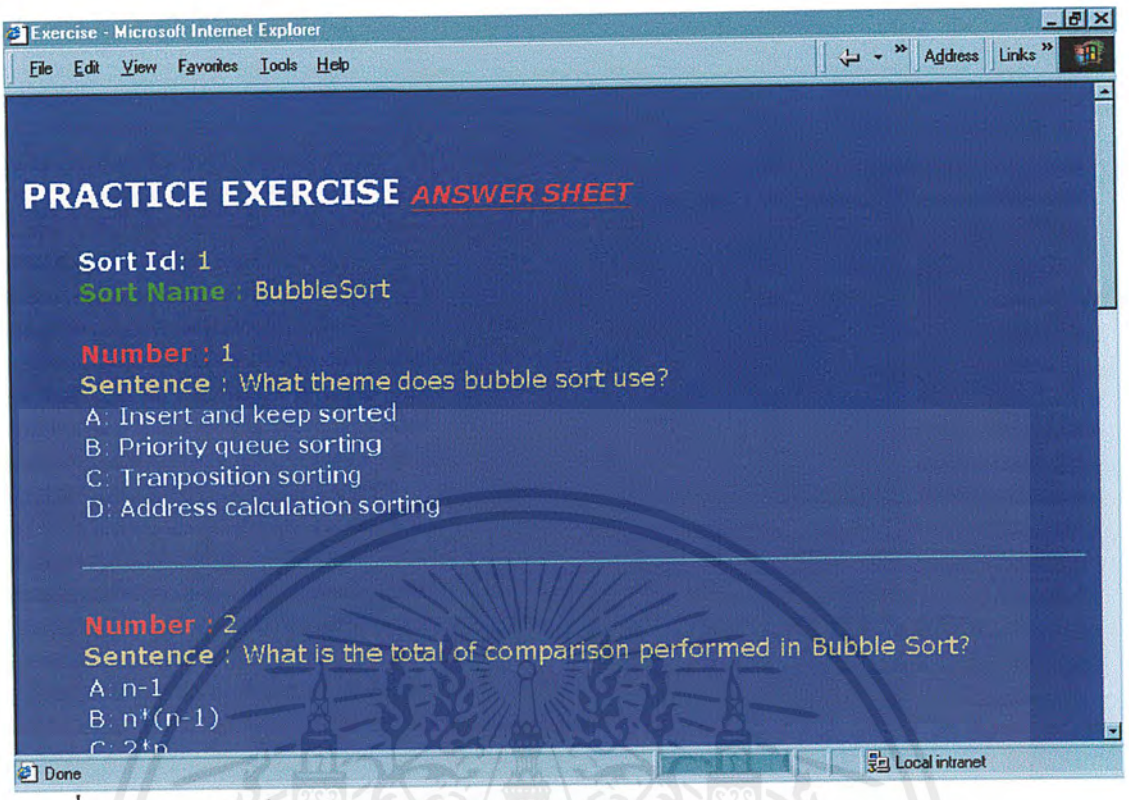


ภาพที่ 4.111 แสดงหน้าจอแสดงส่วนประกอบหลักๆของโปรแกรมแบบฝึกหัด

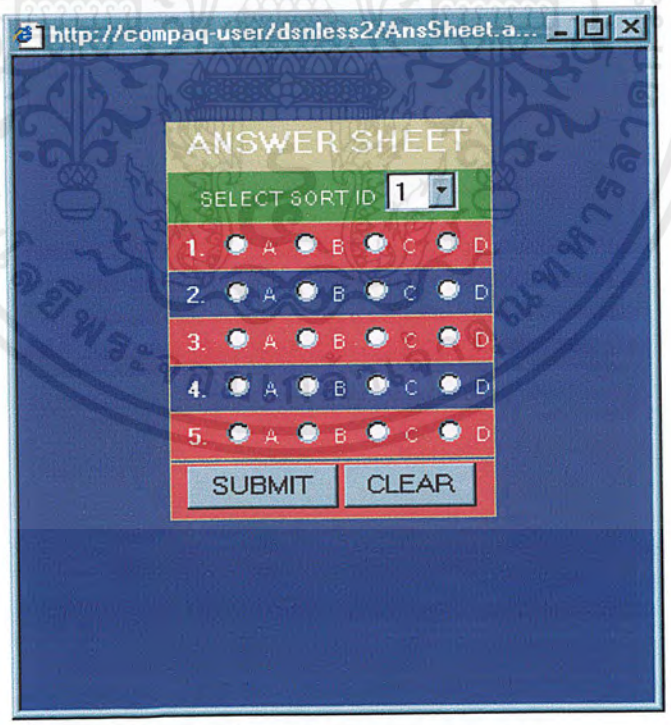


ภาพที่ 4.112 แสดงหน้าจอแสดงส่วนของ Introduction Exercise

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

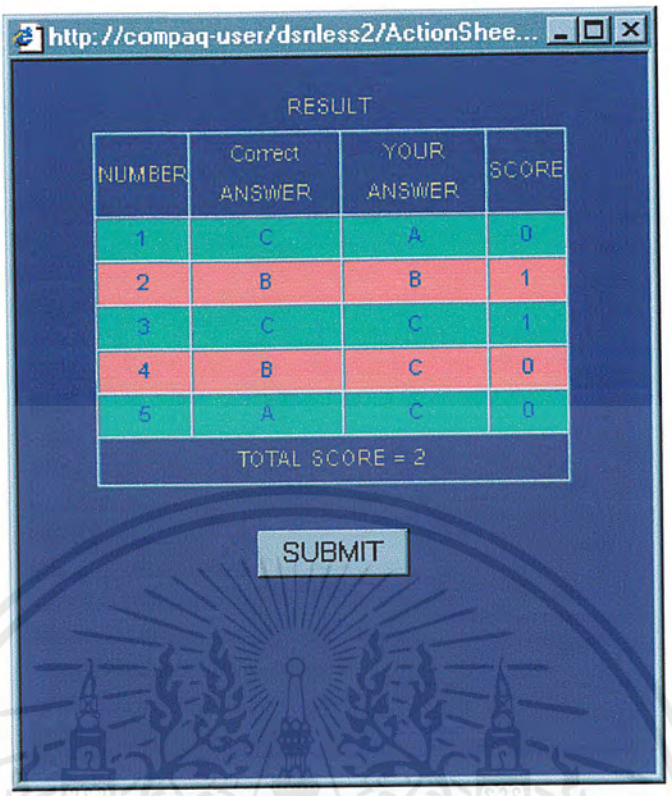


ภาพที่ 4.113 แสดงหน้าจอส่วนของ Practice Exercise

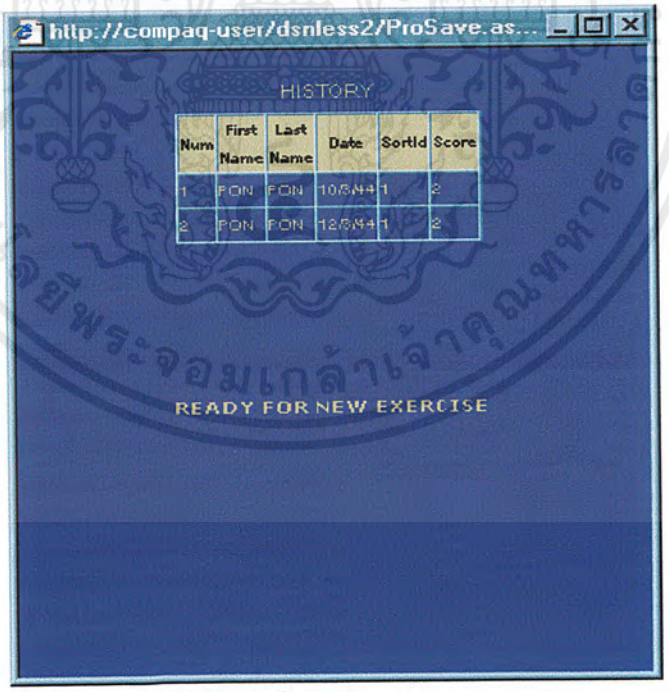


ภาพที่ 4.114 แสดงหน้าจอของ กระดาษคำตอบของPractice Exercise

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

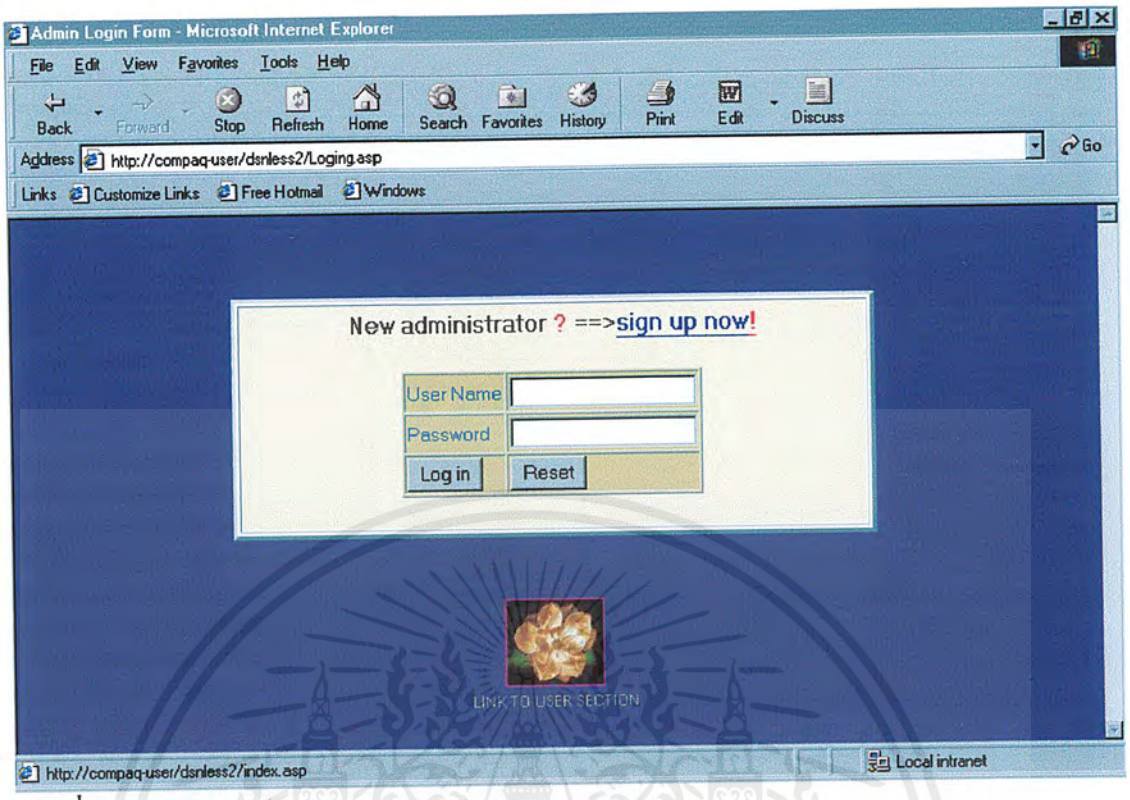


ภาพที่ 4.115 แสดงหน้าจอแสดงผลการทดสอบ

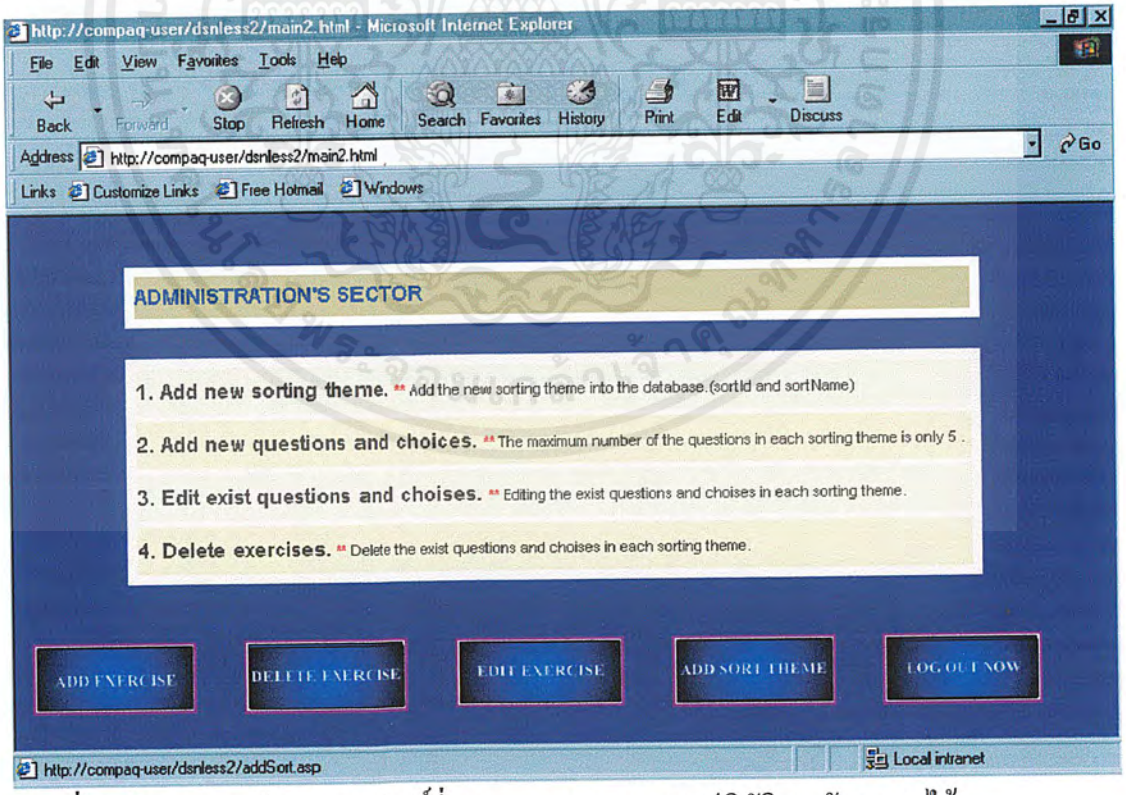


ภาพที่ 4.116 แสดงหน้าจอแสดงประวัติการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.117 แสดงหน้าจอ Log In สำหรับ Administrator



ภาพที่ 4.118 แสดงหน้าจอแสดงสิทธิ์ที่ administrator สามารถปฏิบัติการกับระบบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลในภาพรวม

ในการทำปัญหาพิเศษฉบับนี้ก่อประโยชน์ให้กับผู้ใช้โปรแกรมที่ต้องการศึกษาเนื้อหาและรายละเอียดของการจัดเรียงข้อมูลด้วยวิธีต่างๆอีกทั้งต้องการทำความเข้าใจกับหลักการและขั้นตอนการทำงานของการจัดเรียงข้อมูลแต่ละวิธีให้กระจ่างอย่างเป็นขั้นตอนโดยคณะผู้จัดทำได้จัดทำโปรแกรมให้ผู้ใช้สามารถเลือกใช้งานได้จำนวน 2 ภาษา คือ ภาษาไทยและภาษาอังกฤษ

นอกจากนี้ผู้ใช้ยังสามารถทดสอบความรู้เกี่ยวกับการจัดเรียงข้อมูลด้วยแบบฝึกหัดที่ได้จัดเตรียมไว้อย่างเป็นระบบง่ายและสะดวกต่อการใช้งานอีกทั้งยังมีส่วนติดต่อกับผู้ใช้ที่สวยงามอีกด้วย

5.2 สรุปผลส่วนของโปรแกรมสาธิตการจัดเรียงข้อมูล

โปรแกรมสาธิตการจัดเรียงข้อมูลที่จัดทำขึ้นนี้มีรูปลักษณ์ที่สวยงามน่าใช้และมีความจำเป็นอย่างยิ่งสำหรับผู้ใช้โปรแกรมเพื่อช่วยในการทำความเข้าใจและเห็นภาพการทำงานของแต่ละวิธีการจัดเรียงข้อมูล ได้ชัดเจนและเป็นระบบมากยิ่งขึ้น เพราะเนื่องด้วยความซับซ้อนของอัลกอริทึมซึ่งอาจส่งผลให้ผู้ใช้ไม่สามารถทำความเข้าใจได้ด้วยตนเอง ดังนั้นการจัดทำโปรแกรมสาธิต ซึ่งแสดงการจัดเรียงชุดข้อมูลที่คณะผู้จัดทำได้พิจารณาว่าเหมาะสมที่จะใช้เป็นตัวอย่างสำหรับการเรียงลำดับแต่ละวิธี จึงเป็นประโยชน์และมีคุณค่าอย่างยิ่ง แต่อย่างไรก็ตามชุดข้อมูลตัวอย่างที่คณะผู้จัดทำได้เลือกสรรมานี้ มีเพียง 1 ชุดต่อวิธีการจัดเรียงข้อมูลหนึ่งๆซึ่งอาจไม่เพียงพอในการทำความเข้าใจสำหรับผู้ใช้และอาจไม่ครอบคลุมในทุกๆรูปแบบของข้อมูล

5.3 สรุปผลส่วนของโปรแกรมจำลองการจัดเรียงข้อมูล

โปรแกรมจำลองการจัดเรียงข้อมูลที่จัดทำขึ้นนี้มีส่วนติดต่อกับผู้ใช้ที่สวยงามและน่าใช้เป็นอย่างยิ่ง อีกทั้งส่วนของการใส่ข้อมูลยังสะดวกในการใช้งาน และ มีการทำงานที่ให้ผลลัพธ์ที่ถูกต้องตามอัลกอริทึมของการจัดเรียงข้อมูลแต่ละวิธี แต่ก็มีข้อจำกัดคือไม่สามารถรับค่าของข้อมูลได้มากเนื่องจากการรับค่าเป็นจำนวนมากจะทำให้เกิดความลำบากในการจัดหน้าจอ Animation แสดงผลลัพธ์การจัดเรียงข้อมูลให้มีความสวยงามและนอกจากนี้ความเร็วในการแสดงผลลัพธ์ของการจัดเรียงข้อมูลในแต่ละขั้นตอนนั้นได้ถูกกำหนดขึ้นเป็นค่าตายตัวโดยผู้ใช้ไม่สามารถทำการเลือกหรือเปลี่ยนแปลงความ เร็วในการแสดงผลได้ซึ่งความเร็วในการแสดงผลที่ได้จัดทำขึ้นนี้ เหมาะสำหรับผู้ใช้ที่ต้องการเห็นภาพการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานอย่างชัดเจนและต้องการศึกษาการทำงานอย่างเป็นขั้นตอน ซึ่งเป็นไปตามวัตถุประสงค์ของโปรแกรมที่ได้จัดทำขึ้นเพื่อให้ผู้ใช้ที่มีความสนใจจะศึกษาวิธีการจัดเรียงข้อมูลแต่ละวิธีได้ศึกษาและใช้งานโปรแกรม

5.4 สรุปผลส่วนของโปรแกรมระบบแบบฝึกหัด

ระบบแบบฝึกหัดที่จัดทำขึ้นนี้ มีการจัดการอย่างเป็นระบบทั้งในส่วนของผู้ใช้และส่วนของผู้ดูแลระบบ อีกทั้งมีหน้าจอที่สวยงามน่าใช้งานและสามารถใช้งานผ่านเครือข่ายอินเตอร์เน็ตได้ ดังนั้นในส่วนของแบบฝึกหัดนี้จึงสามารถนำไปประยุกต์ใช้กับการเรียนการสอนวิชาอื่นๆได้ ถ้าได้รับการพัฒนาระบบเพิ่มเติม

5.5 ข้อเสนอแนะ

โปรแกรมจำลองการจัดเรียงข้อมูลผ่านเครือข่ายอินเตอร์เน็ตที่จัดทำขึ้นนี้จะมีประโยชน์มากยิ่งขึ้นหากมีการพัฒนาระบบเพิ่มเติมได้แก่

- การเพิ่มเสียงในการอธิบายเนื้อหาแต่ละขั้นตอน
- การเพิ่มจำนวนชุดตัวอย่างข้อมูลในส่วนของโปรแกรมสาธิตการจัดเรียงข้อมูล
- การแบ่งระดับผู้ใช้ออกเป็นกลุ่มๆเพื่อกำหนดความเร็วในการแสดงผลในส่วนของโปรแกรมจำลองการจัดเรียงข้อมูลให้เหมาะสมกับกลุ่มของผู้ใช้มากยิ่งขึ้น
- การเพิ่มเติมในส่วนของการถามและตอบปัญหาของผู้ใช้ Online

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ไพศาล โมลิสกุลมงคล. 2543 . **พัฒนา Web Database ด้วย ASP**. กรุงเทพ : ไทยเจริญการพิมพ์

ธวัชชัย งามสันติวงศ์ . 2543. **การใช้ Macromedia Flash 4** . ครั้งที่ 1.กรุงเทพ : โรงพิมพ์ บริษัท 21 เซ็นจูรี่ จำกัด

จิตเกษม พัฒนาศิริ . **เสริมแต่งโฮมเพจให้มีชีวิตชีวา ด้วย JAVA SCRIPT** . ครั้งที่ 1 ปี 2541 .กรุงเทพ : โรงพิมพ์ บริษัท วิตตี้ กรุ๊ป จำกัด

สวัสดิ์ ไกรคุ้ม . **การออกแบบเว็บกราฟิกด้วย HTML 4.0** . ครั้งที่ 1 ปี 2541 .กรุงเทพ : เดอะไรบารี พับบลิชิง

กิตติ ภักดีวัฒนะกุล และ ไชยรัตน์ ปานปั้น.2543 . **ASP ฉบับฐานข้อมูล** .ครั้งที่ 1. กรุงเทพ : บริษัท เคที พีคอมพ์ แอน คอนซัลท์

พณรังสี สู่ความดี และ ประชา พฤกษ์ประเสริฐ . 2543 . **สร้างเว็บเพจอย่างไร้ขีดจำกัด ASP เพื่อการประยุกต์ใช้งาน** . กรุงเทพ : บริษัท ซัคเซสมีเดีย

Weiss , Mark Allen . 1992. **Data Structures and algorithm analysis**. Redwood City, Calif : Benjamin/Cummings Publishing

Sartaj Sahni .2000.**Data Structures , algorithms and applications in Java**. Boston : Mc Graw-Hill

Thomas A. Standish .1995. Data Structures , algorithms and software principles in C . Reading,Mass :Addison - Wesley

Rick Darnell .1998 .HTML 4 . Professional reference 1st ed. Indianapolis,Ind : Sams. Net

Eric Vera . 2000 . **Marcromedia Flash5 ActionScript Reference Guide** . 1st ed. San Francisco : Macromedia,Inc .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้