บทที่ 2

การเรียงลำดับข้อมูล

2.1 การเรียงลำดับข้อมูลโดยคอมพิวเตอร์

การเรียงลำดับข้อมูลโดยคอมพิวเตอร์ คือ วิธีการนำเอาข้อมูลมาเรียงต่อกันตามค่าคีย์ของข้อมูลอันหนึ่ง จากค่าน้อยไปค่ามาก หรือ จากค่ามากไปค่าน้อย ข้อมูลจะถูกจัดเรียงลำดับตามค่าของคีย์ข้อมูล ซึ่งถ้าค่าของคีย์ข้อมูลเป็นค่าตัวเลข การเรียงลำดับเรียงตามค่าตัวเลขของคีย์ ส่วนคีย์ที่เป็นชุดตัวอักษร การเรียงลำดับเรียงตามค่าลำดับตัวอักษร (Collating Sequence) ที่เครื่องคอมพิวเตอร์นั้นใช้

การเรียงลำดับข้อมูล แบ่งเป็น 2 ประเภท คือ

1. การเรียงลำดับภายใน (Internal Sorting) เป็นการเรียงลำดับซึ่งข้อมูลทั้งหมดสามารถใส่ลงในหน่วยความจำหลักได้ทั้งหมด การทำงานเกิดขึ้นในหน่วยความจำหลักซึ่งสามารถทำการ Random Access ข้อมูลได้

2. การเรียงลำดับภายนอก (External Sorting) เป็นการเรียงลำดับข้อมูลขนาดใหญ่ซึ่งข้อมูลทั้งหมด ไม่สามารถใส่ลงในหน่วยความจำได้ ต้องเก็บอยู่ในหน่วยความจำสำรองบางส่วน เช่น จานแม่เหล็ก เทปแม่เหล็ก

อัลกอริทึม (Algorithm) ของวิธีการเรียงลำดับข้อมูล ได้มีการพัฒนามาช้านาน ซึ่งก็มีหลายวิธี โดยผู้ต้องการใช้ สามารถเลือกวิธีการทำงาน และนำไปพัฒนาใช้งาน เทคนิควิธีการเรียงลำดับข้อมูลที่นิยมใช้กัน เช่น การเรียงลำดับแบบลอยตัว (Bubble Sort), อินเซอท-ชันซอร์ต (Insertion Sort), ควิกซอร์ต (Quicksort), เชลซอร์ต (Shellsort), ฮีปซอร์ต (Heapsort) แต่ละอัลกอริทึมการเรียงลำดับข้อมูลมีประสิทธิภาพแตกต่างกัน ประสิทธิภาพของการเรียงลำดับก็คือจำนวนเวลาที่ใช้ในการทำงานของอัลกอริทึม (Actual Execution Time of Algorithm) การออกแบบของอัลกอริทึมการเรียงลำดับข้อมูลมีวัตถุประสงค์หลักใหญ่คือเพื่อลดการแลกเปลี่ยนข้อมูล ลดการเคลื่อนย้ายข้อมูล เพราะการสลับตำแหน่งที่อยู่ข้อมูลใช้เวลาในการทำงานมาก

หลักในการพิจารณาเพื่อเลือกวิธีการเรียงลำดับข้อมูลที่จะใช้

1. เวลาที่จะใช้ในการพัฒนาโปรแกรมมาใช้งาน (Programming Time)
2. เวลาที่ใช้ในการทำงานของโปรแกรม (Execution Time of the Program)
3. ขนาดของหน่วยความจำหลัก และหน่วยความจำสำรองที่โปรแกรมต้องการใช้ (Memory and Auxilary Space Need)

พบว่าในบางครั้งนั้น ความพยายามที่จะปรับปรุงประสิทธิภาพของวิธีการเรียงลำดับ จะทำให้โปรแกรมมีความซับซ้อนมากขึ้น (Complexity) และการใช้ตัวแปรมากขึ้น ทำให้ต้องใช้หน่วยความจำในการประมวลผลมากขึ้นเกิดความสิ้นเปลือง (Overhead) มากขึ้นแก่ซอฟต์แวร์ เพราะต้องนำคำสั่งมาและประมวลผลคำสั่งของโปรแกรมที่มีขนาดใหญ่ขึ้น ซึ่งถ้าหากว่า ระบบที่ใช้ (Machine Environment) อนุญาตให้ได้สำหรับขนาดหน่วยความจำที่เพิ่มขึ้นแล้ว การเพิ่มความซับซ้อนของโปรแกรมนั้นทำให้สามารถเพิ่มประสิทธิภาพได้ ก็นับได้ว่าเป็นการแลกเปลี่ยนที่มีค่า (Worth Trade Off)

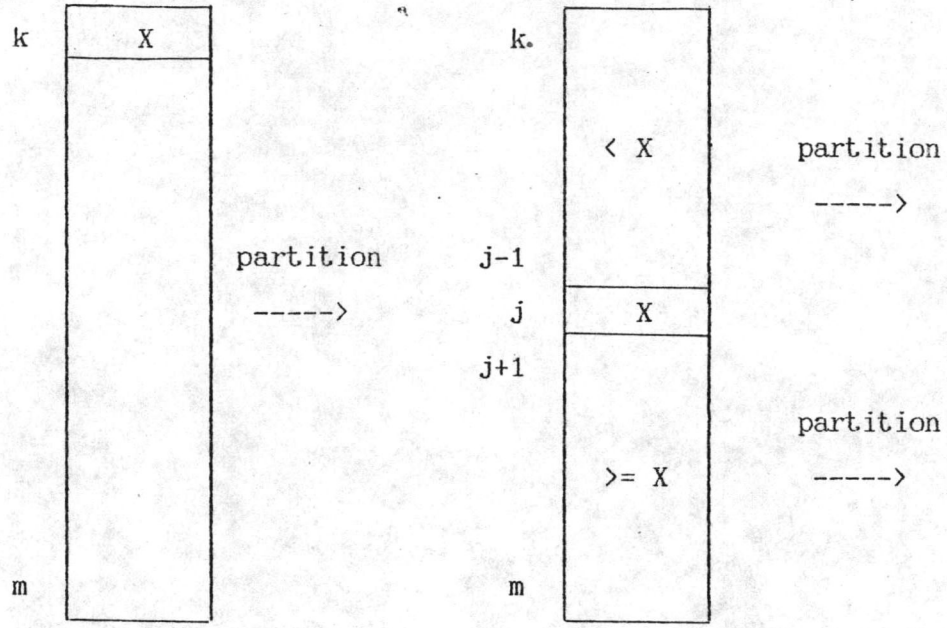
2.2 อัลกอริทึมการเรียงลำดับข้อมูลควิกซอร์ต

ควิกซอร์ต เป็น อัลกอริทึมการเรียงลำดับข้อมูลภายในที่มีประสิทธิภาพและเป็นที่นิยมใช้กันมาก อัลกอริทึมควิกซอร์ต คิดค้นโดย C.A.R Hoare ในปี ค.ศ. 1961

ควิกซอร์ต เป็นเทคนิคการเรียงลำดับที่ได้รับการยอมรับว่ามีประสิทธิภาพ เพราะใช้เวลาการทำงานน้อย และใช้เนื้อที่หน่วยความจำในการประมวลผลน้อย

ลักษณะการทำงานของควิซอร์ต เป็นวิธีการเรียงลำดับข้อมูลที่อาศัยหลักของการแบ่งส่วนข้อมูล (Partition-Exchange Sort) คือในตอนแรกการแบ่งส่วนข้อมูลจะทำให้ชุดลำดับของข้อมูลทั้งหมด ถูกแบ่งออกเป็น 2 ส่วน โดยที่ค่าคีย์ทั้งหมดที่อยู่ในส่วนแรกมีค่าคีย์ข้อมูลน้อยกว่าคีย์ทั้งหมดที่อยู่ในส่วนที่สอง ซึ่งจำนวนคีย์ในแต่ละส่วนไม่จำเป็นต้องเท่ากันต่อไป แต่ละส่วนจะถูกแบ่งออกไปเรื่อยๆ โดยใช้วิธีการเดียวกัน จนกระทั่งค่าคีย์ทั้งหมดได้ถูกเรียงลำดับแล้ว แต่ละขั้นตอนของวิธีการดำเนินงาน จุดประสงค์คือ เพื่อที่จัดวางข้อมูลที่เลือกขึ้นมาในแต่ละขั้นตอนลงในตำแหน่งที่แท้จริงของมัน (คือตำแหน่งที่ข้อมูลจะอยู่เมื่อได้เรียงลำดับเรียบร้อยแล้ว) ซึ่งข้อมูลที่อยู่ก่อนหน้าจะมีคีย์ข้อมูลน้อยกว่า และข้อมูลที่อยู่ตามหลังจะมีค่าคีย์ข้อมูลมากกว่า การทำงานนี้เรียกว่า การแบ่งส่วนข้อมูล (Partition) การแบ่งส่วนนี้จะทำซ้ำกันเรื่อย ๆ กับชุดข้อมูลย่อยเรื่อย ๆ จนกระทั่ง ข้อมูลทั้งหมดถูกวางอยู่ในตำแหน่งที่แท้จริงของมันแล้ว การแบ่งส่วนออก

เป็น 2 ส่วนนี้ช่วยให้การเรียงลำดับเร็วขึ้น เนื่องจากโดยเฉลี่ยแล้วจำนวนคีย์ในแต่ละส่วนที่ได้หลังจากการแบ่งส่วนแล้วจะลดลงครึ่งหนึ่ง ทำให้จำนวนครั้งของการเปรียบเทียบลดลงครึ่งหนึ่งด้วย



รูปที่ 2.1 แสดงการแบ่งส่วนข้อมูล

ข้อเสียของอัลกอริทึมควิกซอร์ตในการนำไปใช้งาน ซึ่งก็นับว่ามีความสำคัญต่อการนำข้อมูลที่ได้เรียงลำดับแล้วไปใช้งาน คือ การทำงานของอัลกอริทึมควิกซอร์ตเป็นประเภท ไม่เสถียรภาพ (Unstable) กล่าวคือ โดยทั่วไปแล้วข้อมูลที่จะทำการเรียงลำดับ ถ้าข้อมูลมีค่าคีย์ข้อมูลเท่ากัน

เช่น ข้อมูลตัวที่ i และ ตัวที่ j มีคีย์ข้อมูลเท่ากัน คือ

key (i) = key (j)

โดย i และ j เป็น ลำดับของข้อมูล ถ้าก่อนการเรียงลำดับข้อมูล ค่า i น้อยกว่าค่า j เมื่อเรียงลำดับข้อมูลจะได้ ข้อมูลตัวที่ i มาก่อน ข้อมูลตัวที่ j การเรียงลำดับที่ให้ผลดังนี้ เป็นการเรียงลำดับที่เสถียรภาพ (Stable) แต่การทำงานของอัลกอริทึมควิกซอร์ต โดยการเปรียบเทียบข้อมูล จะเปรียบเทียบข้อมูลในตำแหน่งที่อยู่ห่างไกลกัน คือ จากตอนปลายทั้งสองด้านของชุดลำดับ ทำให้ผลการเรียงลำดับข้อมูล เป็นไปได้ที่เกิด ไม่เสถียรภาพของข้อมูล ซึ่งไม่สามารถรักษาลำดับเดิมของข้อมูลที่มีค่าคีย์ข้อมูลเท่ากัน

ความเสถียรภาพของข้อมูลนับว่ามีความจำเป็น ถ้าลำดับของข้อมูลก่อนการเรียงลำดับมีความสำคัญในการใช้งาน

2.3 การสำรวจผลงานทางวิชาการ

การพัฒนาปรับปรุงอัลกอริทึมการเรียงลำดับด้วย ควิกซอร์ต ให้มีประสิทธิภาพสูง ได้มีแนวความคิด ตลอดจนผลงานวิชาการ ดังนี้

Motzkin[3] เสนอว่า ข้อเสียเปรียบอย่างหนึ่งของควิกซอร์ต คือการทำงานกับข้อมูลที่อยู่ในลำดับบ้างแล้ว (Partially Sorted) โดยการเลือกตัวกำหนดการแบ่งส่วนข้อมูล จะทำให้เกิดเป็นลักษณะการทำงานที่เป็นกรณีร้ายแรง (Worst Case Situation) และได้เสนอว่า คีย์ที่ควบคุมการแบ่งส่วนควรได้จากค่าเฉลี่ย (Mean) จะทำให้ ลักษณะลำดับของคีย์ ไม่มีผลต่อประสิทธิภาพการทำงานของอัลกอริทึมได้ แต่กรณีร้ายแรง (Worst Case) สามารถเกิดได้เมื่อการกระจายของค่าคีย์ภายหลังการเรียงลำดับแล้ว มีลักษณะดังนี้

$$k_{j-1} < \sum_{i=1}^{j} k_i <= k_j$$

กล่าวคือ ค่าของคีย์ทั้งหมดก่อนตัวที่ j มีค่าอยู่ระหว่างค่าของคีย์ตัวที่ j-1 และ คีย์ตัวที่ j หรืออีกนัยหนึ่งคือ ค่าเฉลี่ยของคีย์ทั้งหมดก่อนคีย์ตัวที่ j มีค่าอยู่ระหว่างค่าคีย์ตัวที่ j-1 และ ค่าคีย์ตัวที่ j ตัวอย่างเช่น ข้อมูล 2 3 6 10 12 23 120 จะเห็นว่าค่าเฉลี่ยของข้อมูลทั้งหมดนั้นคือ 25 ทำให้การใช้ค่าเฉลี่ยของข้อมูลเป็นตัวกำหนดการแบ่งส่วนนั้นทำให้เกิดกรณีร้ายแรงขึ้นได้เช่นกัน

Motzkin[4] ได้เสนอว่า ในการแบ่งส่วนข้อมูล โดยการอ่านข้อมูลไปในทิศทางเดียวกัน แทนการอ่านข้อมูลสลับกันของปลายทั้งสองด้านของข้อมูล เพื่อให้เกิดเสถียรภาพของข้อมูล

Cook และ Do Jin Kim[5] ได้ศึกษาเปรียบเทียบการทำงานของอัลกอริทึม การเรียงลำดับข้อมูล อินเซอทชันซอร์ท, เชลซอร์ต, สเทรท เมอจ ซอร์ต (Straight Merge Sort), ควิกซอร์ต และ ฮีปซอร์ต กับ ข้อมูลที่มีการเรียงลำดับในอัตราส่วนต่างๆ พบว่าโดยการใช้ อินเซอทชันซอร์ต ร่วมกับ ควิกซอร์ต จะทำให้เกิดประสิทธิภาพการทำงานดีขึ้น

---

[3] Motzkin, D., "Meansort," Commun ACM 23, 11(Nov 1980) : 250-251.

[4] Motzkin, D., "A stable Quicksort," Softw. Pract. Exper. 11 (June 1981) : 607-611.

[5] Cook, C. R. , and Do Jin Kim., "Best sorting algorithm for Nearly sorted lists," Commun ACM 26, 4(April 1983) : 620-624.

Baer และ Yi-Bing Lin[๕] เสนอโครงสร้างข้อมูลโคดเวิร์ดเพื่อใช้ในอัลกอริทึมควิกซอร์ต ดังนี้

จากแนวความคิดของโคดเวิร์ด เป็นค่าข้อมูลใหม่ที่ได้จากการนำคีย์ข้อมูลเดิมมาปรับเปลี่ยน โดยค่าข้อมูลใหม่มีคุณสมบัติใช้เปรียบเทียบได้เช่นเดียวกับคีย์ข้อมูลเดิม ในการใช้ค่าข้อมูลใหม่หรือค่าโคดเวิร์ดนี้เปรียบเทียบข้อมูลแทนค่าคีย์ เพื่อการแบ่งส่วนข้อมูลสามารถจัดข้อมูลออกเป็น 4 กลุ่มแทน 2 กลุ่ม คือ กลุ่ม S1, S2, S3 และ S4 โดย ค่าคีย์ข้อมูลในกลุ่ม S1 และ S2 จะเป็นข้อมูลที่มีค่าคีย์น้อยกว่าค่าคีย์ของตัวกำหนดการแบ่งส่วน และค่าคีย์ข้อมูลในกลุ่ม S3 และ S4 จะเป็นข้อมูลที่มีค่าคีย์มากกว่าค่าคีย์ของตัวกำหนดการแบ่งส่วน

โดยการทำงานของอัลกอริทึมควิกซอร์ต ของ Baer และ Yi-Bing Lin ทำตามลักษณะการทำงานเดิมของอัลกอริทึมควิกซอร์ต แต่ใช้ค่าโคดเวิร์ดในการเปรียบเทียบแทนการใช้ค่าคีย์ข้อมูล โดยแบ่งการทำงานออกเป็น 2 ขั้นตอน ขั้นตอนแรกแบ่งข้อมูลทั้งหมดเป็น 2 ส่วน คือ ชุดลำดับบน (T1) และชุดลำดับล่าง (T2) ซึ่ง T1 ประกอบด้วยสมาชิกของ S1 และ S2 ส่วน T2 ประกอบด้วยสมาชิกของ S3 และ S4 โดยแต่ละสมาชิกจะถูกทำเครื่องหมายเพื่อแสดงถึงกลุ่มย่อยที่ตนเองเป็นสมาชิกจริง ๆ การทำงานในขั้นตอนแรกนี้เป็นลักษณะของอัลกอริทึมควิกซอร์ตแบบเดิมที่แบ่งข้อมูลออกเป็น 2 ส่วน ในขั้นตอนที่สอง T1 และ T2 จะต้องถูกอ่านทั้งหมดอีกครั้ง เพื่อกระจายเป็น S1, S2 และ S3, S4 ตามลำดับ โดยการพิจารณาจากขนาดของข้อมูลในแต่ละกลุ่มย่อย ถ้าขนาดของ S1 (S3) และ S2 (S4) มีขนาดเท่า ๆ กันแล้วให้กระจาย T1 และ T2 เรียกว่าการทำโฟร-เว พาทิชชันนิง (Four-Way Partitioning) หรือการแบ่ง 4 ส่วน แต่ถ้าขนาดของ S1 (S3) ต่างจาก S2 (S4) มาก การอ่านข้อมูลอีกรอบเพื่อกระจาย T1, T2 ไม่เป็นประโยชน์เท่าที่ควร ให้ใช้ T1 และ T2 เป็นข้อมูลเข้าในรอบถัดไปแทน เป็นการทำทู-เว พาทิชชันนิง (Two-Way Partitioning) หรือการแบ่ง 2 ส่วน หรือทำเป็น ทรี-เว พาทิชชันนิง (Three-Way Partitioning) การแบ่ง 3 ส่วน ประกอบด้วยกลุ่มข้อมูล S1, S2 และ T2 หรือ กลุ่มข้อมูล T1, S3 และ S4 ในขั้นตอนที่ 2 ทำงานเพื่อพิจารณาการแบ่งข้อมูลต่อให้เป็นข้อมูลกลุ่มย่อย ซึ่งเป็นการแบ่งตามเครื่องหมายที่ให้ไว้ในขั้นตอนแรก

---

[๕]Baer J. L. , and Yi-Bing Lin, "Improving Quicksort Performance with a Codeword Data Structure," IEEE Trans on Software Engineering 15 ,5(May 1989) : 622-631.