บทที่ 2

วิธีการเรียงลำดับข้อมูลทั่วไป

การเรียงลำดับข้อมูลโดยทั่วไปมีหลายวิธี วิธีที่สำคัญมีดังนี้ [1]

- แบบการเปรียบเทียบ (Sorting by comparison)
- แบบแทรก (Sorting by insertion)
- แบบเชลล์ (Shell's sort method)
- แบบลอยตัว (Bubble sort method)
- แบบควิคซอท (Quick sort method)
- แบบเรดิกซ์-เอกซ์เชนจ์ (Radix-exchange sort method)
- แบบฮีพซอท (Heap sort method)
- แบบเลือกแทนที่ (Replacement-selection Sorting)

(ผังงานแสดงวิธีการเรียงลำดับปรากฏในภาคผนวก ก.)

1. แบบการเปรียบเทียบ

1.1 แบบการเปรียบเทียบปกติ วิธีนี้เป็นแบบที่ง่ายที่สุด วิธีการทำคือ การนำเอาค่าคีย์ของแต่ละระเบียนมาเปรียบเทียบกันทุกตัว

---

[1] D.E. Knuth, The Art of Computer Programming Vol. 3, Sorting and Searching (Addison-Wesley Publishing Company, 1973), p. 2.

ขั้นตอนการทำ

กำหนดให้ i,j เป็นตำแหน่งของคีย์ใด ๆ N เป็นจำนวนข้อมูลทั้งหมด

1. ให้ $K_i = K_1$
2. เปรียบเทียบ K_i กับ K_j สำหรับทุกค่า $1 \leq j \leq N$
3. ถ้า K_j ตัวใดมีค่าน้อยกว่า K_i แล้ว ให้นำข้อมูลทั้งสองระเบียนสลับตำแหน่งกัน
4. เพิ่มค่า i อีก 1
5. ทำขั้นที่ 2 จนกระทั่ง i มีค่าเท่ากับ N

1.2 แบบการเปรียบเทียบโดยการนับ วิธีนี้เป็นการเรียงลำดับแบบการเปรียบเทียบผสมกับการนับ

ขั้นตอนการทำ

1. ทำให้ค่าของ COUNT (1)ถึง COUNT(N) มีค่าเป็น 0
2. ให้ i = N
3. ทำขั้นที่ 4 สำหรับค่าของ j = i-1, i-2,--, 1
4. เปรียบเทียบ K_i กับ K_j ถ้า $K_i < K_j$ แล้วให้เพิ่มค่า COUNT(j) ด้วย 1 แต่ถ้า K_i ไม่น้อยกว่า K_j แล้ว ให้เพิ่มค่า COUNT(i) ด้วย 1
5. ลดค่า i ลง 1 ถ้า i มีค่าเท่ากับ 1 หยุดการทำงาน ถ้า i ไม่เท่ากับ 1 กลับไปทำขั้นที่ 3

ข้อสังเกต ขั้นตอนการทำงานนี้จะไม่มีการเคลื่อนย้ายระเบียนใด ๆ ที่ต้องการจัดเรียงลำดับเลย ข้อมูลใน COUNT จะบอกถึงตำแหน่งของระเบียนที่เรียงลำดับแล้วว่าระเบียนใดจะอยู่ก่อนหรือหลัง

2. แบบแทรก

การเรียงลำดับแบบนี้จะถือเสมือนว่ามีระเบียนที่เรียงลำดับอยู่ก่อนแล้ว ดังนั้น การแทรกระเบียนใดเข้าไปในแฟ้มข้อมูล จะกระทำโดยการนำระเบียนนั้นแทรกเข้าไปในตำแหน่งที่ข้อมูลนั้นควรจะอยู่ เพิ่มเข้าไปทีละระเบียนจนกระทั่งหมดทุกระเบียนที่ต้องการนำมาเรียงลำดับ

ขั้นตอนการทำ

กำหนดให้ R เป็นระเบียนเก็บข้อมูลชั่วคราว K เป็นคีย์ชั่วคราว

1. ทำขั้นที่ 2 ถึงขั้นที่ 5 สำหรับค่า j = 2,3,..., N เมื่อ j = N สิ้นสุดการทำงาน

2. ให้ i มีค่าเท่ากับ j-1

ให้นำค่าจาก K_j ไปเก็บที่ K

นำค่าจาก R_j ไปเก็บที่ R

3. เปรียบเทียบค่า K กับ K_i ถ้า K มีค่ามากกว่าหรือเท่ากับ K_i แล้วให้ไปทำขั้นที่ 5

4. นำข้อมูลในระเบียน R_i ไปใส่ที่ระเบียน R_{i+1} แล้วลดค่า i ลง 1 ถ้า i มีค่ามากกว่า 0 ไปทำขั้นที่ 3

5. นำข้อมูลในที่เก็บข้อมูลชั่วคราว R ไปใส่ในระเบียน R_{i+1}

3. แบบเชลล์

การเรียงลำดับแบบเชลล์นี้คิดค้นโดยโดแนล แอล เชล ในปี ค.ศ. 1959 เป็นวิธีการที่ได้ปรับปรุงวิธีการเรียงลำดับแบบ 2 แต่ใช้วิธีการเปรียบเทียบข้ามตำแหน่ง โดยการแบ่งระเบียนออกเป็นกลุ่ม ตัวอย่างเช่น ถ้ามีระเบียนที่ต้องการเรียงลำดับอยู่ 16 ระเบียน ตอนแรกอาจจะแบ่งระเบียนทั้งหมดออกเป็น 8 กลุ่มคือ $(R_1, R_9), (R_2, R_{10}), \ldots, (R_8, R_{16})$ เรียงลำดับแต่ละกลุ่มเหล่านี้ก่อน ต่อมาแบ่งเป็น 4,2,1 กลุ่มตามลำดับ วิธีการแบ่งกลุ่มนี้ไม่มี

ข้อจำกัด คือจะแบ่งเป็น h_t, h_{t-1},---, h_1 ใด ๆ ก็ได้ แต่ h_1 ต้องมีค่าเท่ากับ 1 เสมอ วิธีการแบ่งที่ดีที่สุดคือ[1]

ให้ $h_1 = 1$, $h_{s+1} = 3h_s+1$ และหยุดแบ่งที่ h_t เมื่อ $h_{t+2} \geqslant N$

ขั้นตอนการทำ

กำหนดให้ h_t, h_{t-1},---, h_1 เป็นลำดับของตัวเลขที่ใช้ในการแบ่งกลุ่ม โดยที่ $h_1 = 1$

1. ทำขั้นที่ 2 สำหรับค่า s = t, t-1,---, 1 แล้วจึงจบขั้นตอนการเรียงลำดับนี้

2. ให้ h เท่ากับ h_s แล้วทำขั้นที่ 3 ถึง 6 สำหรับค่า $h < j \leqslant N$

3. ให้ i มีค่าเท่ากับ j-h

   ให้ K มีค่าเท่ากับ K_j

   ให้ R มีค่าเท่ากับ R_j (นำ R_j ไปเก็บไว้ที่ R ซึ่งเป็นที่เก็บข้อมูลชั่วคราว)

4. ถ้า $K \geqslant K_i$ ไปทำขั้นที่ 6

5. นำข้อมูลใน R_i ไปใส่ที่ R_{i+h} แล้วเอาค่า i-h ไปใส่ที่ i ถ้า i มีค่ามากกว่า 0 กลับไปทำขั้นที่ 4

6. นำข้อมูลใน R ไปใส่ที่ R_{i+h}

4. แบบลอยตัว

เป็นวิธีการเรียงลำดับโดยการเปลี่ยนที่ระหว่างระเบียนที่มีค่าคีย์มากกับระเบียนที่มีค่าคีย์น้อย ระเบียนที่มีค่าคีย์มากสุดจะค่อย ๆ เคลื่อนไปทางขวาจนถึงตำแหน่งสุดท้ายที่ไม่มี

---

[1] D.E. Knuth, The Art of Computer Programming vol. 3, Sorting and Searching (Addison-Wesley Publishing Company, 1973), p. 95.

ระเบียนใดต่อไปอีก ระเบียนนั้นก็จะเป็นระเบียนที่ R_N ใช้วิธีการเดิมกับระเบียนที่เหลือจะได้ระเบียนที่ R_{N-1} ทำเช่นนี้จนได้ R_1 ซึ่งแสดงว่าขั้นตอนการเรียงลำดับเสร็จสิ้นลงแล้ว

ขั้นตอนการทำ

1. ให้ตัวแปร BOUND มีค่าเท่ากับ N

2. ให้ t มีค่าเท่ากับ 0 แล้วทำขั้นที่ 3 สำหรับ j = 1,2,..., BOUND-1 เสร็จแล้วไปทำขั้นที่ 4

3. ถ้า $K_j > K_{j+1}$ สลับที่ระเบียน R_j กับ R_{j+1} และให้ t มีค่าเท่ากับ j

4. ถ้า t = 0 แสดงว่าขั้นตอนการเรียงลำดับเสร็จสิ้นลงแล้ว

   ถ้า $t \neq 0$ ให้นำค่า t ไปเก็บที่ BOUND แล้วกลับไปทำขั้นที่ 2

5. แบบควิคซอท

ปรับปรุงมาจากแบบ 4 แต่ใช้ผลจากการเปรียบเทียบในครั้งก่อน เพื่อตัดสินว่าจะเปรียบเทียบคีย์ใดต่อไป

แนวความคิดของวิธีนี้คือ ครั้งแรกจะนำระเบียน 1 ระเบียน สมมุติเรียกว่า R_1 เคลื่อนไปยังตำแหน่ง S ที่ R_1 ควรอยู่ ในขณะที่หาตำแหน่ง S นั้น ก็จัดระเบียนทางด้านซ้ายของ S ให้มีค่าน้อยกว่าหรือเท่ากับ S และระเบียนทางขวามีค่ามากกว่าหรือเท่ากับ S ดังนั้นแฟ้มข้อมูลจะถูกแบ่งออกเป็นแฟ้มข้อมูลย่อยคือ $R_1, R_2, \ldots R_{S-1}$ และ $R_{S+1}, \ldots, R_N$ ใช้วิธีการเดิมกับแต่ละแฟ้มข้อมูลย่อยจนได้ข้อมูลที่เรียงลำดับกัน

วิธีที่จะนำมากล่าวนี้เป็นแนวความคิดของอาร์ เซดวิค (R. Sedgewick) โดยใช้ตัวชี้ i และ j ให้ i = 2 และ j = N ถ้าพบ R_i ใด ๆ ที่อยู่ทางด้านซ้ายจะเพิ่มค่า i ขึ้น 1 และเปรียบเทียบต่อไปจนพบ R_i ที่อยู่ทางด้านขวา ทำวิธีการแบบเดียวกันกับค่าของ j จนพบ R_j ที่ควรจะอยู่ทางด้านซ้ายเปรียบเทียบค่า i และ j ถ้า $i < j$ ให้เปลี่ยนข้อมูลในระเบียน R_i และ R_j ทำวิธีการเดิมจนพบ $i \geq j$

ขั้นตอนการทำ ในขั้นตอนการทำนี้ จะปรับปรุงเพื่อเพิ่มประสิทธิภาพดังนี้คือ

1. สมมุติว่ามีคีย์ K_o ซึ่งเท่ากับ $-\infty$ และมีคีย์ $K_{N+1} = +\infty$ โดยที่

$$K_o \leq K_i \leq K_{N+1}$$ สำหรับทุกค่า i ซึ่ง $1 \leq i \leq N$

2. แฟ้มข้อมูลย่อยที่มีจำนวนระเบียนเท่ากับหรือน้อยกว่า M จะนำมาเรียงลำดับแบบแทรกในตอนท้าย เพราะจะทำให้เครื่องเสียเวลาในการทำงานน้อยที่สุด (M = 9 เหมาะสมที่สุด[1])

3. ระเบียนที่มีค่าคีย์เท่ากัน จะสลับตำแหน่งกัน (ตามแนวความคิดของ อาร์ ซี ชิงเกิลตันจะทำให้การทำงานเร็วขึ้น[2])

หลังจากแก้ไขแล้ว สามารถสรุปขั้นตอนการทำงานโดยให้ค่า r ℓ บอกตำแหน่งคีย์ดังนี้คือ

1. ถ้า $N \leq M$ ทำขั้นที่ 9 มิฉะนั้นให้สะแตคว่าง ให้ $\ell = 1$ และ $r = N$

2. ให้ $i = \ell$ $j = r + 1$ $K = K_\ell$ และนำข้อมูลใน R_ℓ ไปเก็บที่ที่เก็บระเบียนชั่วคราว R

3. เพิ่มค่า i อีก 1 ถ้า K_i น้อยกว่า K แล้ว ทำขั้นตอนนี้ซ้ำ

4. ลดค่า j ลง 1 ถ้า K_j มากกว่า K แล้ว ทำขั้นตอนนี้ซ้ำ

5. เปรียบเทียบค่า i กับ j ถ้า $j \leq i$ สลับข้อมูลระเบียน R_ℓ กับ R_j แล้วไปทำขั้นที่ 7

6. สลับข้อมูลระเบียน R_i กับ R_j ไปทำขั้นที่ 3

7. ถ้า $r-j \geqslant j-\ell > M$ ใส่ค่า (j+1,r) ที่ส่วนบนของสะแตค ให้ r=j-1 ไปทำขั้นที่ 2

ถ้า $j-\ell > r-j > M$ ใส่ค่า $(\ell, j-1)$ ที่ส่วนบนของสะแตค ให้ ℓ=j+1 ไปทำขั้นที่ 2

---

[1] D.E. Knuth, The Art of Computer Vol. 3, Sorting and Searching (Addison-Wesley Publishing Company, 1973), pp. 119-122.

[2] Ibid.

ถ้า $r-j \geq M > j-\ell$ ให้ $\ell = j+1$ ไปทำขั้นที่ 2

ถ้า $j-\ell > M \geq r-j$ ให้ $r = j-1$ ไปทำขั้นที่ 2

8.ถ้าสะแตคไม่ว่าง นำค่าส่วนบนของสะแตคออกมาคือ (ℓ', r') ให้ $\ell = \ell'$ และ $r=r'$ ไปทำขั้นที่ 2

9. สำหรับค่า $j = 2,3,\ldots,N$ ถ้า $K_{j-1} > K_j$ ให้ทำดังนี้คือ

ให้ $K = K_j$ นำข้อมูลใน R_j ใส่ที่ R $i = j-1$ นำข้อมูลใน R_i ใส่ที่ R_{i+1} $i = i-1$

ทำซ้ำจนกระทั่ง $K_i \leq K$ แล้วใส่ข้อมูลระเบียน R ลงในระเบียน R_{i+1}

6. <u>แบบเรดิกซ์-เอกซ์เชนจ์</u>

จะเปลี่ยนค่าคีย์ทุกตัวให้เก็บอยู่ในรูปของเลขฐาน 2 การเปรียบเทียบจะทำทีละบิท ว่าเป็น 0 หรือ 1 ซึ่งมีหลักการทำงานคล้ายแบบที่ 5

<u>ขั้นตอนการทำ</u> สมมุติคีย์ที่นำมาเรียงลำดับแทนได้ด้วยเลขฐาน 2 $(a_1 a_2 \ldots a_m)$ a_i จะเรียกว่า บิทที่ i ของคีย์ สะแตคที่จะใช้ต้องเตรียมไว้สำหรับเก็บข้อมูล m-1 ตัว

1. ทำให้สะแตคว่าง และให้ค่า $\ell = 1$ $r = N$ $b = 1$
2. ถ้า ℓ มีค่าเท่ากับ r ไปทำขั้นที่ 10

ถ้า ℓ ไม่เท่ากับ r ให้ i มีค่าเท่ากับ ℓ, j มีค่าเท่ากับ r

3. ตรวจสอบบิทที่ b ของ K_i ถ้ามีค่าเท่ากับ 1 ไปทำขั้นที่ 5
4. เพิ่มค่า i อีก 1 ถ้า $i \leq j$ กลับไปทำขั้นที่ 3 ถ้ามากกว่าไปทำขั้นที่ 8
5. ตรวจสอบบิทที่ b ของ K_j ถ้ามีค่าเท่ากับ 0 ไปทำขั้นที่ 7
6. ลดค่า j ลง 1 ถ้า $i \leq j$ ไปทำขั้นที่ 5 ถ้ามากกว่าไปทำขั้นที่ 8
7. สลับที่ R_i กับ R_j แล้วกลับไปทำขั้นที่ 4
8. เพิ่มค่า b อีก 1 ถ้า b มีค่ามากกว่า m (m คือจำนวนบิททั้งหมดที่เกิดจากการเปลี่ยนค่าคีย์เป็นเลขฐาน 2) ไปทำขั้นที่ 10

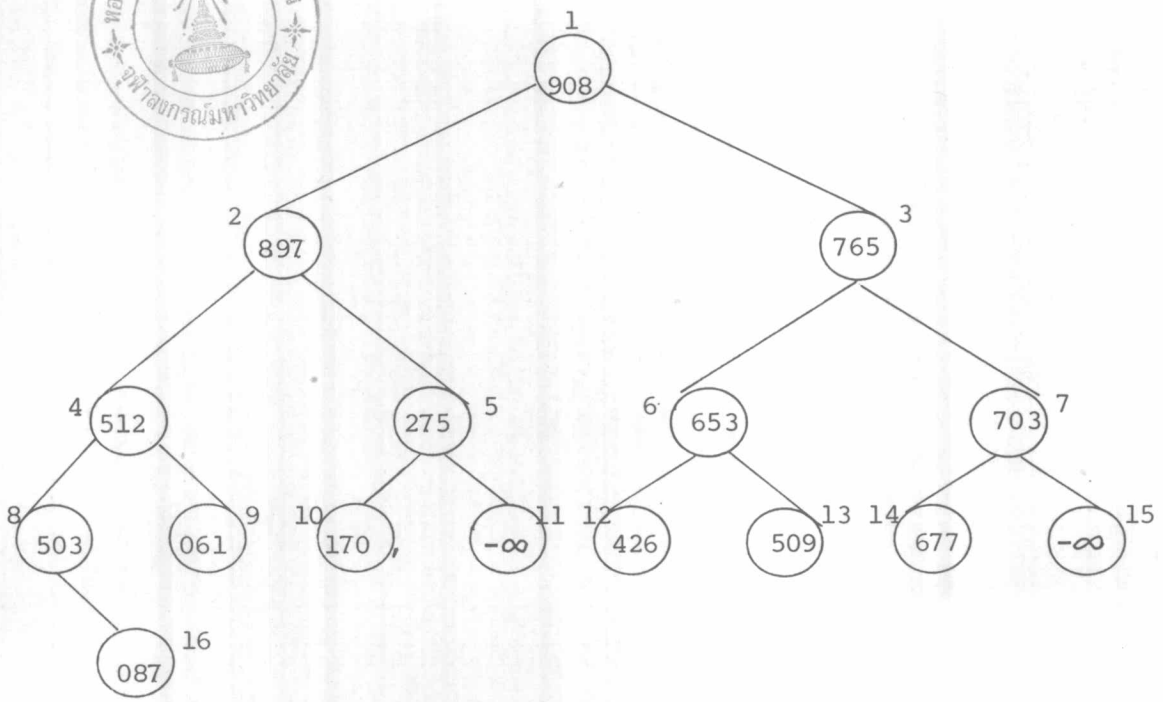
ถ้า b ไม่มากกว่า m เปรียบเทียบค่า j กับ ℓ ถ้า j มีค่าน้อยกว่า ℓ หรือ j = r กลับไปทำขั้นที่ 2 มิฉะนั้นเปรียบเทียบค่า j กับ ℓ อีกครั้ง ถ้า j = ℓ เพิ่มค่า ℓ อีก 1 กลับไปทำขั้นที่ 2

9. ใส่ค่า (r,b) ที่ส่วนบน ของสะแตค และให้ r มีค่าเท่ากับ j กลับไปทำขั้นที่ 2

10. ถ้าสะแตคว่าง แสดงว่าการเรียงลำดับสิ้นสุดลง ถ้าสะแตคไม่ว่าง ให้ค่า ℓ มีค่าเท่ากับ r + 1 และนำค่าทอปของสะแตคออกมา (r′,b) แล้วกำหนดให้ r = r′, b = b′ และกลับไปทำขั้นที่ 2

7. <u>แบบฮีพซอท</u>

แฟ้มข้อมูลที่ประกอบด้วยคีย์ $K_1, K_2, \ldots\ldots, K_N$ จะเรียกว่า "ฮีพ" ถ้า $K_{j/2} \geq K_j$ สำหรับทุก ๆ ค่าของ $1 < j/2 < j \leq N$ ดังนั้นจะได้ว่า $K_1 \geq K_2$, $K_1 \geq K_3$, $K_2 \geq K_4$....เป็นต้น ซึ่งจะทำให้ได้ว่า K_1 จะเป็นคีย์ที่มีค่ามากที่สุด และอยู่บนทอปของฮีพ ทำให้สามารถจัดเรียงลำดับได้ดังตัวอย่างในแผนภาพของฮีพซึ่งจะแสดงให้เห็นดังนี้คือ



รูปที่ 2.1 ทรีของการเรียงลำดับแบบฮีพซอท

จะเห็นว่าถ้าสามารถเปลี่ยนรูปแบบของแฟ้มข้อมูลให้อยู่ในรูปฮีพ จะเรียงลำดับจากด้านบนของทรีลงมาจนถึงด้านล่างของทรีได้

ในการกำหนดคุณสมบัติของฮีพนี้ เจ ดับบลิวเจ วิลเลียม เป็นผู้คิดค้นในปี ค.ศ. 1964 และในปีเดียวกันนั้นเอง อาร์ ดับบลิวฟลอยด์ก็ได้ปรับปรุงการเรียงลำดับแบบฮีพดังนี้คือ สมมุติว่ามีแฟ้มข้อมูลซึ่งจัดเรียงลำดับจนกระทั่งได้ว่า

$K[j/2] \geq Kj$ สำหรับทุกค่าของ $\ell < [j/2] < j \leq N$ ซึ่ง ℓ มีค่ามากกว่าหรือเท่ากับ 1 แล้ว จะสามารถลดค่าของ ℓ ลงจนกระทั่ง ℓ มีค่าเท่ากับ 1 ทำให้ได้ทรีซึ่งอยู่ในรูปของฮีพ จากแนวความคิดนี้เอง ทำให้การเรียงลำดับแบบฮีพเริ่มต้นจากคีย์ตัวใดก่อนก็ได้ (จากขั้นตอนการทำจะเลือกคีย์ตัวที่ N/2)

ขั้นตอนการทำ สมมุติ $N \geq 2$ R เป็นที่เก็บระเบียนชั่วคราว

1. ให้ ℓ มีค่าเท่ากับ $N/2+1$ และ r เท่ากับ N

2. ถ้า $\ell > 1$ ให้ ℓ มีค่าเท่ากับ $\ell-1$

   R มีค่าเท่ากับ R_ℓ

   K มีค่าเท่ากับ K_ℓ

   ถ้า ℓ ไม่มากกว่า 1 ให้ R มีค่าเท่ากับ R_r

   K มีค่าเท่ากับ K_r

   R_r มีค่าเท่ากับ R_1

   r มีค่าเท่ากับ r-1 ถ้า r = 1 ให้ R_1 เท่ากับ R และสิ้นสุดการเรียงลำดับ

3. ให้ j เท่ากับ ℓ

4. ให้ i เท่ากับ j

   j เท่ากับ 2j

   ถ้า $j < r$ ไปทำขั้นที่ 5

   $j = r$ ไปทำขั้นที่ 6

   $j > r$ ไปทำขั้นที่ 8

5. ถ้า $K_j < K_{j+1}$ ให้เพิ่มค่า j อีก 1

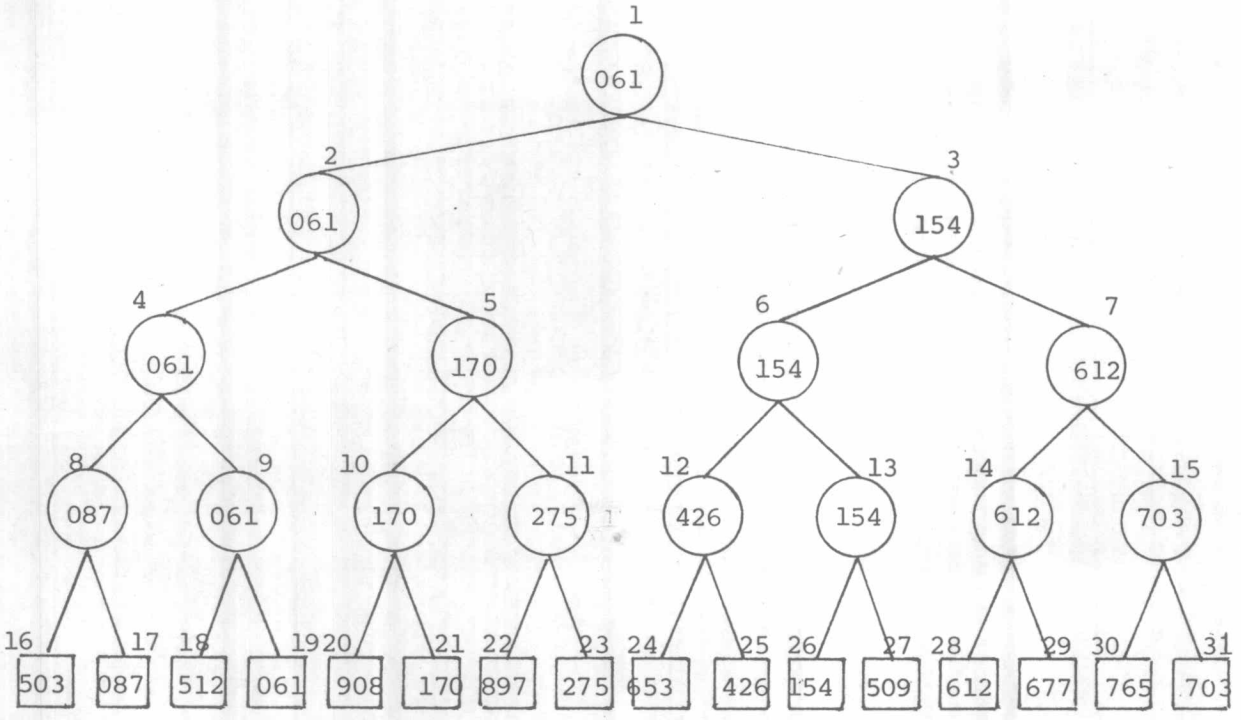
6. ถ้า $K \geq K_i$ ไปทำขั้นที่ 8

7. ให้นำ R_j ไปเก็บที่ R_i และกลับไปทำขั้นที่ 4

8. ให้นำ R ไปเก็บที่ R_i และไปทำขั้นที่ 2

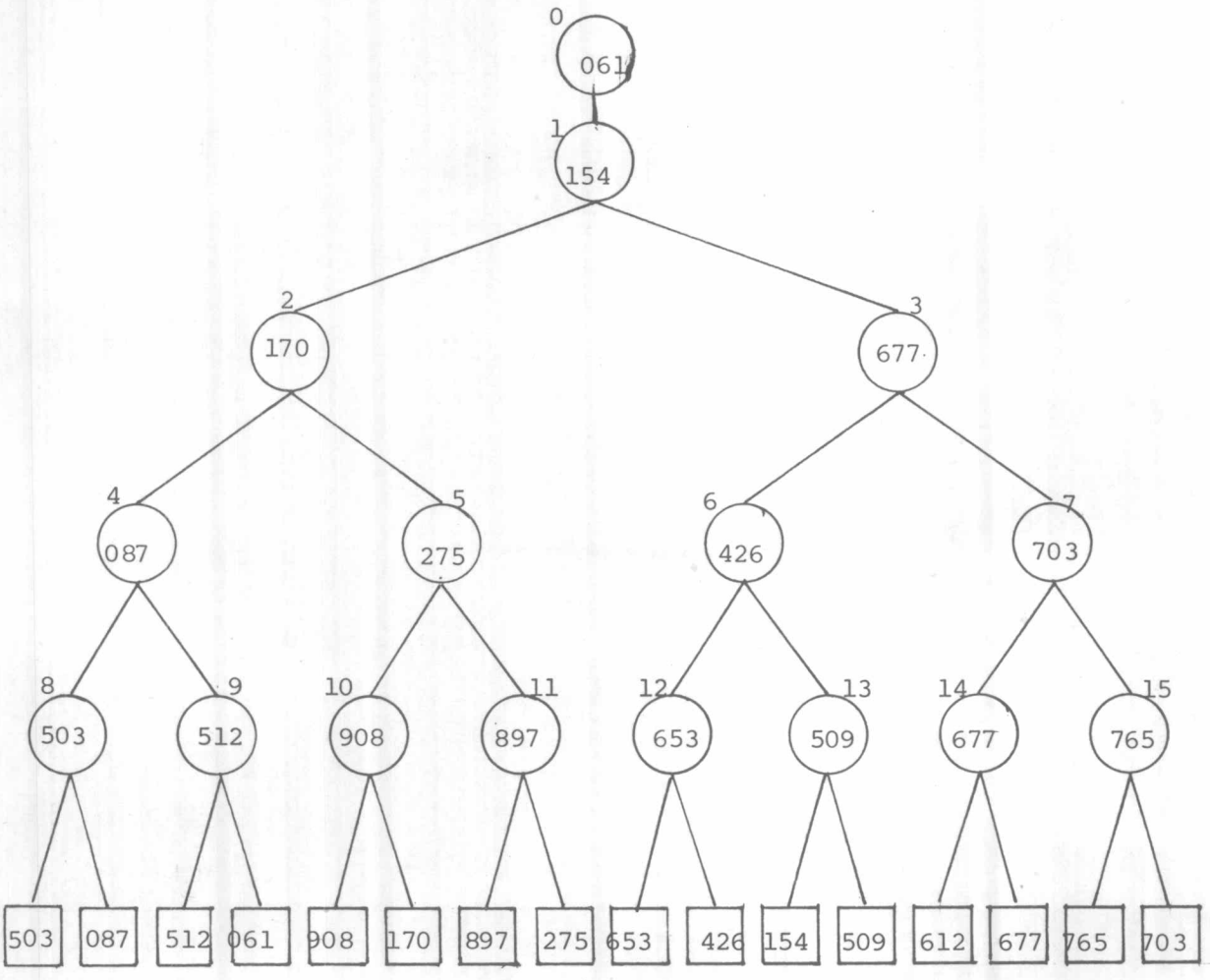
8. <u>แบบเลือกแทนที่</u> เป็นการเรียงลำดับภายนอกหน่วยความจำ คือจำนวนระเบียนที่จะนำมาเรียงลำดับมีมาก จึงแบ่งระเบียนออกมาเป็นส่วนเพื่อให้สามารถเรียงลำดับภายในหน่วยความจำได้ นำส่วนระเบียนเหล่านั้นไปเก็บบนอุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราวได้แก่ เทปแม่เหล็ก หรือจานแม่เหล็ก ดังนั้น วิธีนี้จึงแตกต่างจากวิธีต่าง ๆ ที่กล่าวมา เพราะอุปกรณ์ทำงานได้ช้ากว่าการทำงานภายในหน่วยความจำ

ระเบียนแต่ละส่วนที่เรียงลำดับแล้ว เรียกว่า สตริง (String) การสร้างสตริงจะใช้ทรีคือนำคีย์ของ 2 ระเบียนมาเปรียบเทียบกัน ตัวที่น้อยกว่าเรียกว่า วินเนอร์ (เรียงจากมากไปน้อย) จนกระทั่งได้วินเนอร์ที่มีค่าน้อยที่สุด ดังรูปที่ 2.2



รูปที่ 2.2 แสดงทรีของวินเนอร์

คีย์ 061 จะถูกเคลื่อนออกจากทรี คีย์ที่จะถูกเคลื่อนต่อไปจะเลือกจากคีย์ 512 087 และ 154 ซึ่งไม่อยู่ node ภายใน (node ที่อยู่ภายในวงกลม) ดังนั้น การปฏิบัติงานจริงจึงนิยมเก็บทรีของ ลูสเซอร์ (LOSER) ดังนี้คือ



รูปที่ 2.3 แสดงทรีของลูสเซอร์

วิธีที่จะกล่าวนี้ เป็นของจอห์น อาร์ วอลเตอร์ เจมส์ เพนเตอร์ และมาร์ติน ซอค ซึ่งมีขั้นตอนที่ดีในการสร้างทรี และแยกสตริงออกจากกัน หลักการคือ จะถือเสมือนว่าระเบียนประกอบด้วยคีย์ 2 ตัว (S,K) โดยที่ S เป็นลำดับที่ของสตริง และ K คือค่าของคีย์ ให้ S เป็นคีย์หลัก K เป็นคีย์รอง

โครงสร้างของทรีจะมีลักษณะดังนี้คือ ให้ P เป็นจำนวน node ที่ต้องการ แต่ละ node ใช้เนื้อที่ในการเก็บ c เวิร์ด ดังนั้น node ที่ j จะอยู่บริเวณ $L_o + cj$ ด้วยเขตข้อมูลต่าง ๆ ดังนี้

KEY	คือค่าคีย์ของ node ภายใน
RECORD	คือข้อมูลของระเบียนของ node ภายใน (ค่าคีย์เป็นส่วนหนึ่งในระเบียน)
LOSER	คือตัวชี้ ชี้ไปยัง LOSER ของ node ภายใน
RN	คือลำดับที่ของสตริงของระเบียนที่ LOSER ชี้ไป
FE	คือตัวชี้ ชี้ไปยัง node ภายในที่อยู่เหนือ node ภายนอกนี้
FI	คือตัวชี้ชี้ไปยัง node ภายในที่อยู่เหนือ node ภายในนี้

FE และ FI จะมีค่าคงที่เสมอ โดยปกติจะใช้การคำนวณมากกว่าการเก็บค่าในหน่วยความจำ

<u>ขั้นตอนการทำ</u> จะอ่านแฟ้มข้อมูลนำเข้าและเขียนข้อมูลที่ออกจากทรีอย่างเรียงลำดับ จำนวนสตริงสูงสุดจะเก็บที่ RMAX และ $P \geq 2$ node ให้ X[0], ...,X[P-1] แทนแต่ละ node ซึ่งประกอบด้วยเขตข้อมูลดังกล่าวข้างต้น

1. ให้ RMAX = 0 RC = 0 LASTKEY $= \infty$ Q = LOC (X[0]) และ RQ=0
   สำหรับทุกค่า j ที่ $0 \leq j < P$ สร้างค่าเริ่มต้นแก่ X [j] เมื่อ J = LOC(X [j]) ดังนี้คือ

   LOSER (J) = J RN(J) = 0

   FE(J) = LOC (X[$\lfloor$ (P+j)/2$\rfloor$])

   FI(J) = LOC (X[$\lfloor$j/2$\rfloor$])

2. ถ้า RQ = RC ไปทำขั้นที่ 3 ถ้า RQ > RMAX หยุดการทำงาน หรือมิฉะนั้น ให้ RC = RQ

3. ถ้า RQ ≠ 0 เขียนระเบียน Q และให้ LASTKEY = KEY(Q)

4. ถ้าข้อมูลในแฟ้มข้อมูลหมด ให้ RQ = RMAX + 1 ไปทำขั้นที่ 5
ถ้าไม่หมดอ่านระเบียนใหม่ไปเก็บที่ระเบียน RECORD(Q) เปรียบเทียบ KEY(Q) กับ LASTKEY ถ้า KEY(Q) < LASTKEY ให้ RQ = RQ + 1 และถ้า RQ > RMAX ให้ RMAX = RQ

5. ให้ T = FE(Q)

6. ถ้า RN(T) < RQ หรือถ้า RN(T) = Q และ KEY(LOSER(T)) < KEY(Q) แล้วเปลี่ยนข้อมูลระหว่าง LOSER(T) กับ Q และ RN(T) กับ RQ

7. ถ้า T = LOC(X[1]) แล้ว ไปทำขั้นที่ 2 มิฉะนั้นให้ T = FI(T) ไปทำขั้นที่ 6

นอกจากนี้มีวิธีการเรียงลำดับแบบอื่น ๆ คือ การนับโดยการกระจาย (Distribution Counting) แบบคำนวนตำแหน่ง (Address Calculation Sorting) แบบแบทเชอร์พาราเลล (Batcher's Parallel method) การเรียงลำดับโดยการเมิร์จ ซึ่งแบ่งออกเป็นการเมิร์จแบบสองทาง (Straight two-way merge Sort) แบบลิสท์ (List merge sort) และการเรียงลำดับโดยการกระจาย ได้แก่เรดิกซ์ลิสซ์ซอท (Radix List sort) แบบฮุคกิงอัพออฟคิว (Hooking-up of queues) เป็นต้น