



External (Merge)

Natural, Balanced, Polyphase

เสนอ

อาจารย์สุรีย์พัชร มุสิกะภาวัต

จัดทำโดย

63102105112 นายอัศรพล พิกุลศรี

63102105136 นายสิทธิพร วงศ์บาตร

63102105140 นายชลสิทธิ์ สีสถาน

63102105141 นายธนวัฒน์ สารินทร์

รายงานนี้เป็นส่วนหนึ่งของรายวิชาโครงสร้างข้อมูลและอัลกอริทึม

(14122305)

ภาคเรียนที่ 1 ปีการศึกษา 2564

มหาวิทยาลัยราชภัฏสกลนคร

คำนำ

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชา โครงสร้างข้อมูลและอัลกอริทึม เพื่อให้ได้ศึกษาความรู้เกี่ยวกับ การเรียงลำดับภายนอก (External Sort) ซึ่งมีหลายวิธีดังนี้ Natural merge, Balanced merge และ Polyphase merge

คณะผู้จัดทำหวังว่ารายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือผู้ที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อเสนอแนะหรือข้อผิดพลาดประการใด คณะผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

สารบัญ

| เรื่อง | หน้า |
|-----------------------------------|------|
| หน้าปก | ก |
| คำนำ | ข |
| สารบัญ | ค |
| External Sort | 1 |
| Natural merge sort | 1 |
| การเรียงลำดับแบบผสาน (Merge sort) | 1 |
| ภาพเพิ่มเติมสำหรับ Merge sort | 3 |
| Balanced merge sort | 5 |
| Iterative 2-way merge | 6 |
| Polyphase merge | 7 |
| Polyphase merge example | 7 |
| อ้างอิง | 8 |

External Sort

การเรียงลำดับภายนอก คือ อัลกอริธึมการเรียงลำดับที่สามารถจัดการข้อมูลจำนวนมากได้ เมื่อข้อมูลจัดเรียงไม่พอดี (fit) กับหน่วยความจำหลักของคอมพิวเตอร์ (RAM) และจะต้องอยู่ในหน่วยความจำภายนอกที่ช้ากว่า (Hard drive) แทน

การเรียงลำดับภายนอกมักใช้อัลกอริทึมแบบไฮบริดในการเรียงลำดับส่วนของข้อมูล ที่มีขนาดเล็ก ข้อมูลจะถูกใส่ลงในหน่วยความจำหลัก จะถูกอ่าน จัดเรียง และเขียนลงในไฟล์ชั่วคราว ซึ่งในขั้นตอนการรวมไฟล์ย่อยที่เรียงลำดับจะถูกรวมเป็นไฟล์ใหญ่ไฟล์เดียว

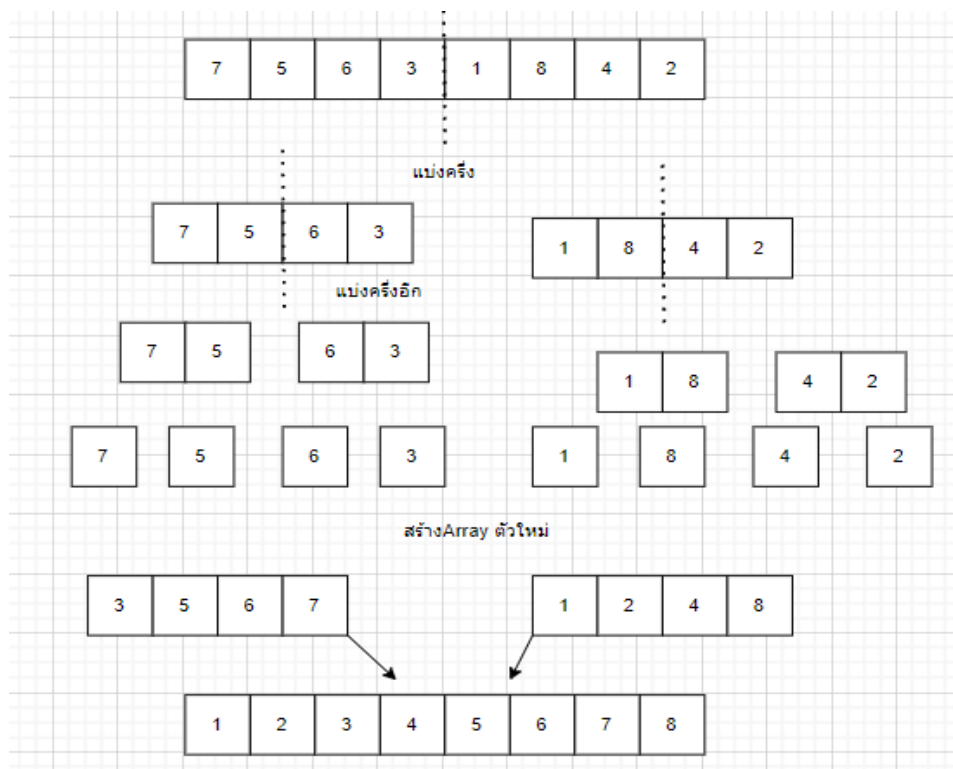
Natural merge sort

การเรียงลำดับแบบผสาน (Merge sort)

“แบ่งรายการที่ไม่เรียงลำดับออกเป็น n รายการย่อย โดยแต่ละรายการมีองค์ประกอบหนึ่งรายการ ซึ่งรายการขององค์ประกอบหนึ่งรายการจะถือเป็นการจัดเรียง รวมรายการย่อยซ้ำๆ เพื่อสร้างรายการย่อยที่เรียงลำดับใหม่จนกว่าจะมีรายการย่อยเหลือเพียงรายการเดียว”

กล่าวคือ รายการเรียงลำดับ merge sort จะเป็นลักษณะ การตัด array ออกเป็น 2 ส่วนในแต่ละส่วนก็จะเอาไป recursion ตัดออกเป็นชั้นย่อย ๆ ลงไปอีกจนเหลือขนาดเล็ก ที่สามารถ sort ได้ ก็จะจัดการ sort ขึ้นเล็กๆ ให้เสร็จแล้วค่อยนำขึ้นเล็กๆ ที่ sort เสร็จมาต่อกันอีกที

การที่จะนำมาต่อกัน เรา จะใช้การวนซ้ำ ซึ่งจะใช้วิธีใดก็ได้ ในที่นี้เราจะใช้ while loop ทำการวนซ้ำ จนกว่าข้อมูล จะหมด โดยแต่ละรอบให้ทำการเช็คค่า ตำแหน่งแรกของ array ย่อย 2 ตัวอันไหนที่น้อยกว่า ให้ใส่ใน array ใหญ่ ดังภาพ

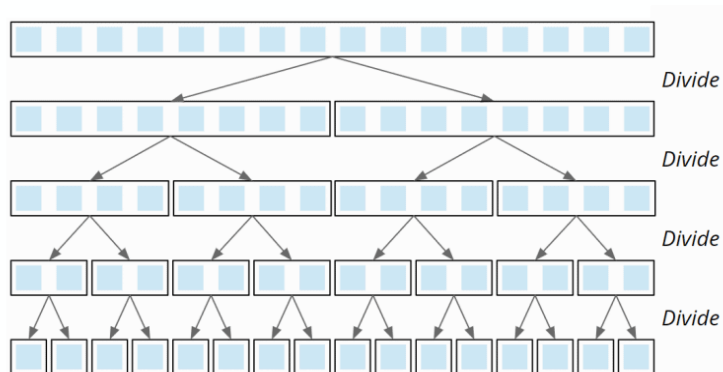


จะเห็นได้ว่า merge sort จะเป็นอัลกอริทึมในลักษณะ recursion และ ใช้หลักการแบ่ง (divide & conquer) ซึ่ง Divide and Conquer เป็นวิธีการแก้ปัญหาแบบเรียกซ้ำ ซึ่งแบ่งปัญหา ออกเป็นปัญหาย่อยที่มีขนาดเล็กลง แก้ปัญหาย่อยแบบเรียกซ้ำ และสุดท้ายรวมการแก้ปัญหาย่อย เพื่อแก้ปัญหาเดิม วิธีนี้มักจะช่วยให้เราลดความซับซ้อนของเวลาได้มาก

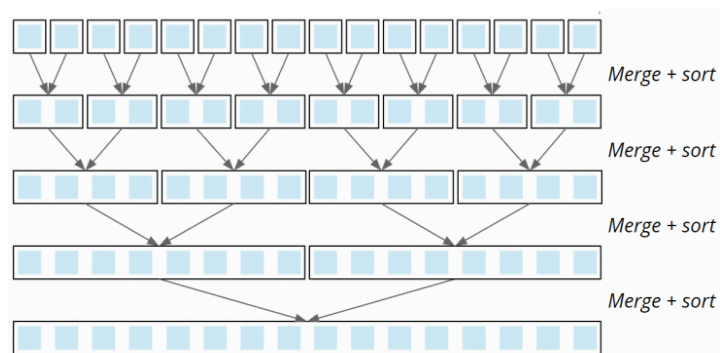
1. Algorithm merge sort (low, height)
2. if (low < heigh):
3. mid = (low + height)/2
4. merge sort(low, mid)
5. merge sort(mid + 1, height)
6. merge (low, mid, height)
7. end

ภาพเพิ่มเติมสำหรับ Merge sort

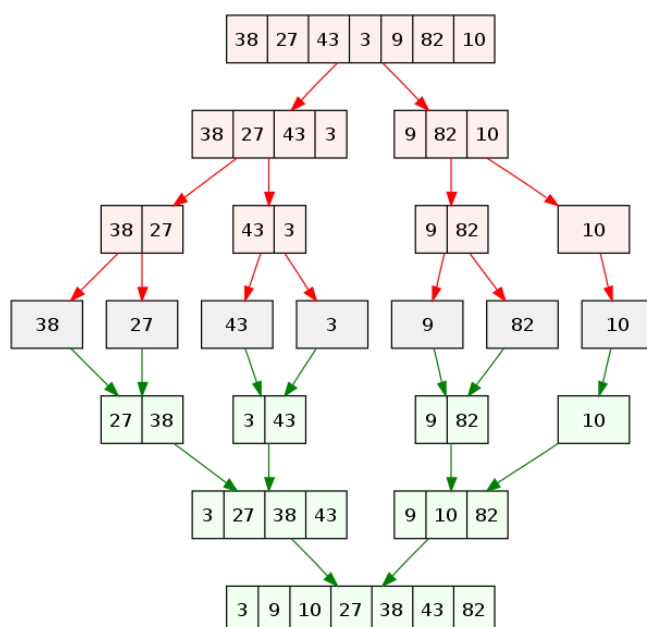
กระบวนการแยก



กระบวนการ merge



ภาพรวม



การ Merge ลิสต์ที่เรียงลำดับแล้วสองลิสต์ รวมกันเป็น ลิสต์เรียงลำดับ ลิสต์เดียว

(two sorted list merge to single sorted list)

| A | B | C | A | B | C |
|----|----|---|----|----|----|
| 2 | 5 | | 2 | 5 | 2 |
| 8 | 9 | | 8 | 9 | 5 |
| 15 | 12 | | 15 | 12 | 8 |
| 18 | 17 | | 18 | 17 | 9 |
| | | | | | 12 |
| | | | | | 15 |
| | | | | | 18 |

| A | B | C |
|----|----|----|
| 2 | 5 | 2 |
| 8 | 9 | 5 |
| 15 | 12 | 8 |
| 18 | 17 | 9 |
| | | 12 |
| | | 15 |
| | | 18 |

m

n

Merge two sorted list
To single sorted list

$\Theta(M+n)$

```

Algorithm Merge(A, B, m, n)
{
    i = 1; j = 1; k = 1;
    while( i <= m & j <= n)
        if (A[i] < B[j])
            C[k++] = A[i++];
        else
            C[k++] = B[j++];
    }
    for( ; i <= m ; i++)
        C[k++] = A[i];
    for( ; j <= n ; j++)
        C[k++] = B[j];

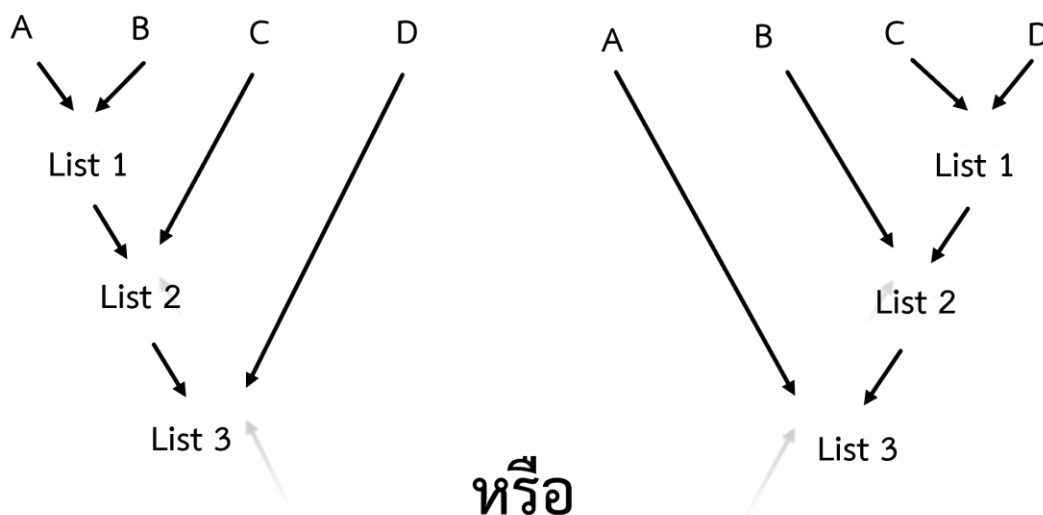
```

Balanced merge sort

การเรียงลำดับการผสาน k-way แบบสมดุลที่จัดเรียงสตริมข้อมูลโดยใช้การผสานซ้ำๆ มันกระจายอินพุตออกเป็นสองสตริมโดยอ่านบล็อกอินพุตที่พอดีกับหน่วยความจำซ้ำๆ รัน เรียงลำดับ แล้วเขียนไปยังสตริมถัดไป จากนั้นจะผสานสองสตริมซ้ำๆ และทำให้แต่ละรันที่ผสานเข้าเป็นหนึ่งในสองเอาต์พุตสตริม จนกว่าจะมีเอาต์พุตที่จัดเรียงเพียงรายการเดียว

ยกตัวอย่าง แม้ว่าดิสก์ไดรฟ์ส่วนใหญ่สามารถทำงานกับไฟล์ชั่วคราวจำนวนมากได้แต่ก็ใช้ไม่ได้กับพื้นที่จัดเก็บเทป ซึ่งเป็นเรื่องยากที่จะมีไฟล์หลายไฟล์ในเทปเดียว และเรามีเทปไดรฟ์ในจำนวนจำกัดเราสามารถทำได้ด้วย 3 หรือ 4 เทป แต่เราสามารถเพิ่มประสิทธิภาพได้ด้วยเทปไดรฟ์มากขึ้น

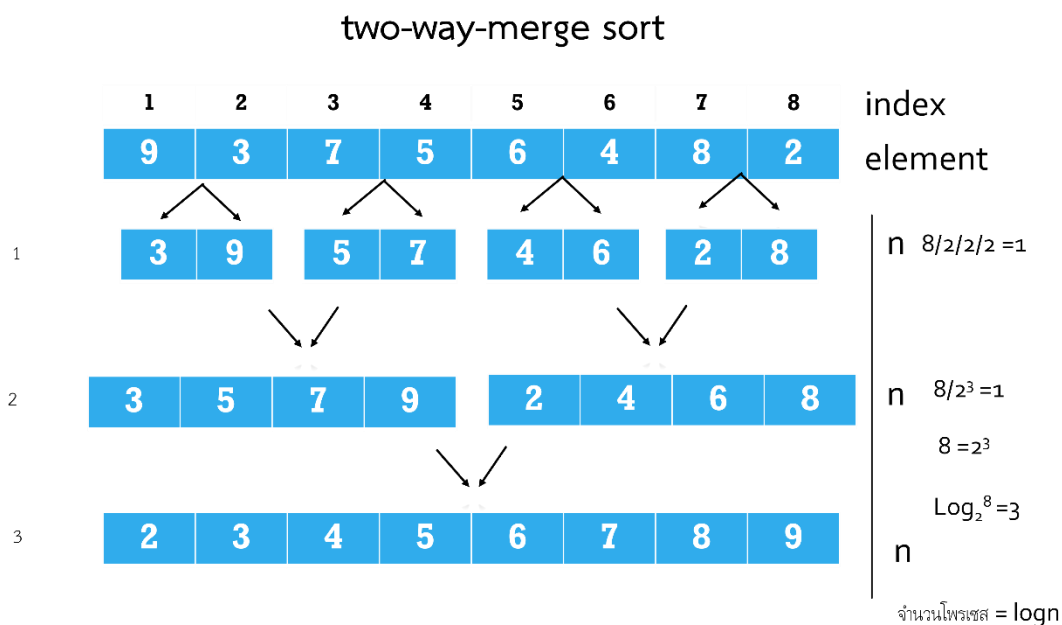
ยกตัวอย่างแบบ 4-way merge



Iterative 2-way merge

การผสาน 2 ทางแบบวนซ้ำ เป็นการผสานอาร์เรย์ k สองชุดซ้ำๆ โดยใช้การผสานแบบ 2 ทางจนเหลือเพียงอาร์เรย์เดียว หากอาร์เรย์ถูกรวมในลำดับที่ต้องการ เวลาทำงานที่ได้จะเป็น $O(kn)$ เวลาทำงานสามารถปรับปรุงได้โดยการรวมครั้งแรกกับครั้งที่สอง ครั้งที่สามกับครั้งที่สี่ และอื่นๆ

เนื่องจาก จำนวนอาร์เรย์ลดลงครึ่งหนึ่งในการวนซ้ำแต่ละครั้ง ทุกองค์ประกอบจะถูกย้ายเพียงครั้งเดียว เวลาทำงานต่อการวนซ้ำจึงอยู่ใน $\Theta(n)$ เนื่องจาก n คือจำนวนขององค์ประกอบ เวลาทำงานทั้งหมดจึงเป็น $\Theta(n \log k)$



Polyphase merge

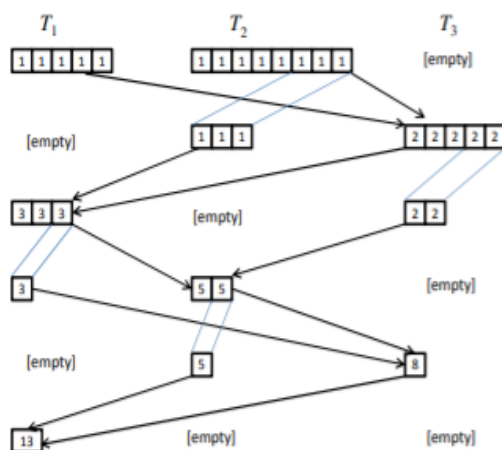
การเรียงลำดับแบบผสานเฟสเป็นรูปแบบของการเรียงลำดับการผสานจากล่างขึ้นบน ที่เรียงลำดับรายการโดยใช้การกระจาย รายการย่อย (runs) ที่ไม่สม่ำเสมอ

ซึ่งส่วนใหญ่ใช้สำหรับการเรียงลำดับภายนอกและมีประสิทธิภาพมากกว่าการเรียงลำดับผสานทั่วไปเมื่อมีน้อยกว่า 8 ไฟล์การทำงานภายนอก (เช่น เทปไดรฟ์หรือไฟล์บนฮาร์ดไดรฟ์) การเรียงลำดับการรวมหลายเฟสไม่ใช้การเรียงลำดับที่เสถียร

Polyphase merge example

สมมติว่าเรามี 3 เทป (T_1 , T_2 , T_3) และเราจะ merge ตามลำดับดังนี้
 จัดเรียงและแจกจ่ายบันทึกไปยัง T_1 และ T_2
 Merge T_1 และ T_2 ไปยัง T_3 เหลือไว้บางตัวใน T_2
 Merge T_2 และ T_3 ไปยัง T_1 เหลือไว้บางตัวใน T_2
 Merge T_3 และ T_1 ไปยัง T_2 เหลือไว้บางตัวใน T_3
 Merge T_1 และ T_2 ไปยัง T_3 เหลือไว้บางตัวใน T_2 และอื่นๆ
 เรามักทิ้งเทปต้นฉบับไว้สองอันและเทปหนึ่งอันสำหรับวางไฟล์ที่ผสาน

Polyphase Merge On 13 Runs



อ้างอิง

Aman Chauhan 1(2560) . polyphase merge sort. ค้นคืนแล้ว 29 กันยายน 2564 แหล่งที่มา
<https://practice.geeksforgeeks.org/problems/explain-poly-way-merge-sort>

CSC 344 – Algorithms and Complexity. Lecture #4 – External Sorting. ค้นคืนแล้ว 29
 กันยายน 2564. แหล่งที่มา <https://home.adelphi.edu/~siegfried/cs344/344l4.pdf>

GeeksforGeeks(2564). Merge Sort. ค้นคืนแล้ว 28 กันยายน 2564. แหล่งที่มา
<https://www.geeksforgeeks.org/merge-sort/>

GeeksforGeeks(2564). External Sorting. ค้นคืนแล้ว 27 กันยายน 2564. แหล่งที่มา
<https://www.geeksforgeeks.org/external-sorting/>

Minsoo Jeon and Dongseung Kim. Load-Balanced Parallel Merge Sort.
 ค้นคืนแล้ว 29 กันยายน 2564. แหล่งที่มา
https://www.researchgate.net/publication/220091378_Parallel_Merge_Sort_with_Load_Balancing#pf4

Son-klin Limthongkul(2557). External sorting. ค้นคืนแล้ว 28 กันยายน 2564. แหล่งที่มา
<https://slideplayer.in.th/slide/2180294/>

Sven Woltmann(2563). Merge Sort – Algorithm, Source Code, Time Complexity.
 ค้นคืนแล้ว 28 กันยายน 2564. แหล่งที่มา
https://www.happycoders.eu/algorithms/merge-sort/#Natural_Merge_Sort

Ta(2012). [ซีทสรุป พี่ต้า] Data Structure & Algorithm. Merge Sort. ค้นคืนแล้ว 28 กันยายน 2564. แหล่งที่มา

https://docs.google.com/file/d/0B09a_TYwhKDZZXNSRHV3aFhuamc/edit?resourcekey=0-JzOfoiKxvZreJUR5U0vPYg

Wikipedia(2564). Merge sort. ค้นคืนแล้ว 27 กันยายน 2564. แหล่งที่มา

https://en.wikipedia.org/wiki/Merge_sort

Wikipedia(2564).Two-way_merge. ค้นคืนแล้ว 29 กันยายน 2564

แหล่งที่มา https://en.wikipedia.org/wiki/K-way_merge_algorithm#Two-way_merge

Wikipedia(2021). Polyphase merge sort. ค้นคืนแล้ว 29 กันยายน 2564. แหล่งที่มา

https://en.wikipedia.org/wiki/Polyphase_merge_sort