

**External (Merge)**

**Natural, Balanced, Polyphase**

# **เสนอ**

**อาจารย์สุรีย์พัชร มุสิกะภวัต**

# **จัดทำโดย**

**63102105112 นายอัครพล พิกุลศรี**

**63102105136 นายสิทธิพร วงศ์บาตร**

**63102105140 นายชลสิทธิ์ สีสถาน**

**63102105141 นายธนวัฒน์ สารินทร์**

# **รายงานนี้เป็นส่วนหนึ่งของรายวิชาโครงสร้างข้อมูลและอัลกอริทึม (14122305)**

**ภาคเรียนที่ 1 ปีการศึกษา 2564**

**มหาวิทยาลัยราชภัฏสกลนคร**

# **คำนำ**

รายงานเล่มนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของวิชา โครงสร้างข้อมูลและอัลกอริทึม (Data Structures and Algorithms) เพื่อให้ได้ศึกษาความรู้เกี่ยวกับ การเรียงลำดับภายนอก (External) มีหลายวิธีดังนี้ Natural merge, Balanced merge และ Polyphase merge

คณะผู้จัดทำหวังว่ารายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หรือผู้ที่กำลังหาข้อมูลเรื่องนี้อยู่ หากมีข้อแนะนำหรือข้อผิดพลาดประการใด คณะผู้จัดทำขอน้อมรับไว้และขอภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

# **สารบัญ**

เรื่อง หน้า

หน้าปก ก

คำนำ ข

สารบัญ ค

External Sorting 1

* การเรียงลำดับภายนอก
* External merge sort algorithm
* การเรียงลำดับแบบผสาน (Merge sort)
* Algorithm merge k sorted arrays

Natural merge 7

* Natural Merge Sort
* ตัวอย่าง Natural Mergesort

Balance merge 8

* Two-way merge

- k-way merge คือ

- two-way merge

- Iterative 2-way merge

Polyphase merge 11

* Polyphase merge example

อ้างอิง 12

# **1. External Sorting**

## **1.1 การเรียงลำดับภายนอก**

## การเรียงลำดับภายนอก คือ อัลกอรึธึมการเรียงลำดับที่สามารถจัดการข้อมูล จำนวนมหาศาลได้ เมื่อข้อมูลจัดเรียงไม่พอดี (fit) กับหน่วยความจำหลักของคอมพิวเตอร์ (RAM) และจะต้องอยู่ในหน่วยความจำภายนอกที่ช้ากว่า (Hard drive) แทน

การเรียงลำดับภายนอกมักใช้อัลกอริธึมแบบไฮบริด ในการเรียงลำดับส่วนของข้อมูลที่มีขนาดเล็กพอที่จะใส่ลงในหน่วยความจำหลักจะถูกอ่าน จัดเรียง และเขียนลงในไฟล์ชั่วคราว ในขั้นตอนการรวม ไฟล์ย่อยที่เรียงลำดับจะถูกรวมเป็นไฟล์เดียวที่ใหญ่กว่า

## **1.2 External merge sort algorithm**

## อัลกอริธึมการเรียงลำดับการผสานภายนอก เป็นตัวอย่างหนึ่งของการเรียงลำดับภายนอก ซึ่งจัดเรียงชิ้นส่วนที่แต่ละชิ้นพอดีใน RAM แล้วนำชิ้นที่เรียงมารวมกัน

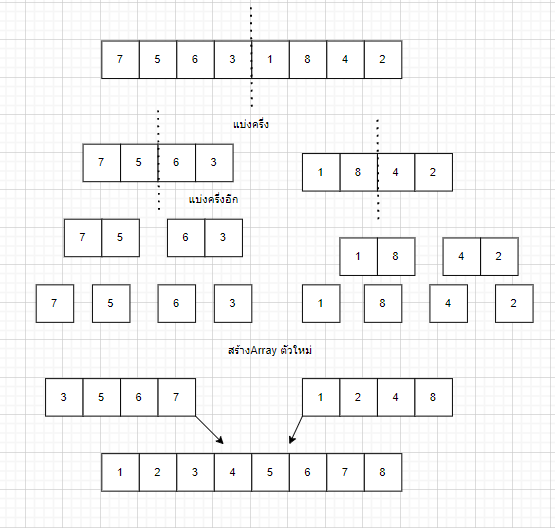
## อันดับแรกเราจะทำการแบ่งไฟล์ออกเป็น ส่วนๆ เพื่อให้ขนาดของไฟล์รันมีขนาดเล็กพอที่จะใส่ลงในหน่วยความจำหลักได้ จากนั้นเรียงลำดับการทำงานแต่ละครั้งในหน่วยความจำหลักโดยใช้อัลกอริทึมการเรียงลำดับแบบผสาน ในที่สุดก็รวมการทำงานที่เป็นผลลัพธ์เข้าด้วยกันเป็นการรันที่ใหญ่ขึ้นเรื่อย ๆ จนกว่าไฟล์จะถูกจัดเรียง

## **1.3 การเรียงลำดับแบบผสาน (Merge sort)**

1. แบ่งรายการที่ไม่เรียงลำดับออกเป็น n รายการย่อย โดยแต่ละรายการมีองค์ประกอบหนึ่งรายการ (รายการขององค์ประกอบหนึ่งรายการจะถือเป็นการจัดเรียง)
2. รวมรายการย่อยซ้ำๆ เพื่อสร้างรายการย่อยที่เรียงลำดับใหม่จนกว่าจะมีรายการย่อยเหลือเพียงรายการเดียว นี่จะเป็นรายการเรียงลำดับ

merge sort จะเป็นลักษณะ การตัด array ออกเป็น 2 ส่วนในแต่ละส่วนก็จะเอาไป recursion ตัดออกเป็นชิ้นย่อย ๆ ลงไปอีกจนเหลือขนาดเล็ก ที่สามารถ sortได้ ก็จะจัดการ sort ชิ้นเล็ก ๆ ให้เสร็จ แล้วค่อยนำ ชิ้นเล็ก ๆ ที่ sort เสร็จมาต่อกันอีกที

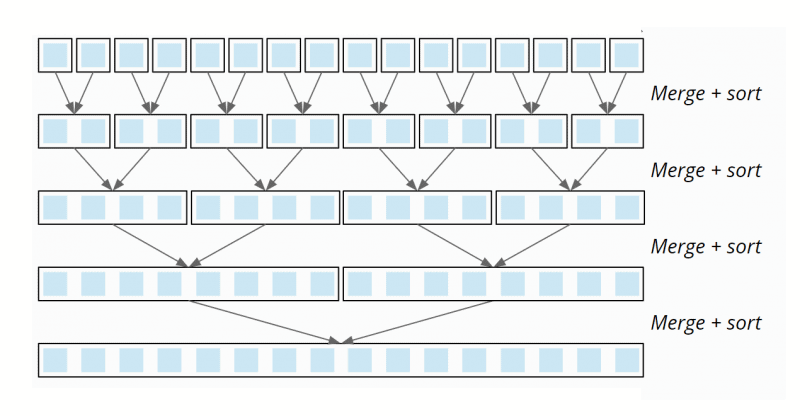
การที่จะนำมาต่อกัน เราจะใช้การวนซ้ำ ซึ่งจะใช้วิธีใดก็ได้ ในที่นี้เราจะใช้ while loop ทำการวนซ้ำ จนกว่าข้อมูล จะหมด โดยแต่ละรอบให้ทำการเช็คว่า ตำแหน่งแรกของ array ย่อย 2 ตัวอันไหนที่น้อยกว่า ให้ใส่ใน array ใหญ่ ดังภาพ



## **ภาพเพิ่มเติมสำหรับ Merge sort**

## 

ref: https://www.happycoders.eu/algorithms/merge-sort/



## **1.4 Algorithm merge k sorted arrays**

1. อ่าน input\_file เพื่อให้องค์ประกอบ 'run\_size' ส่วนใหญ่อ่านได้ในแต่ละครั้ง

ติดตามทุกครั้งที่อ่านในอาร์เรย์

2.เรียงลำดับการทำงานโดยใช้ Merge Sort

3.เก็บอาร์เรย์ที่เรียงลำดับในไฟล์ แทน i สำหรับไฟล์ แต่ละไฟล์

4.รวมไฟล์ที่เรียงลำดับโดยใช้วิธีการที่กล่าวถึง ผสาน k sorted arrays

# **2. Natural merge**

เป็นการผสานครั้งละ 2 input file เรียกว่า Two-way merge ถ้าผสานครั้งละ M input file เรียกว่า M-way merge

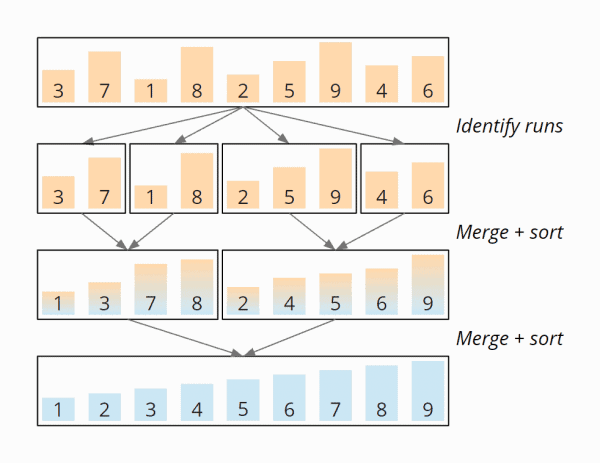
## **2.1 Natural Merge Sort**

Natural Merge Sort คือ การปรับให้เวลาเหมาะสมของ Merge Sort ซึ่งระบุพื้นที่ ที่จัดเรียงไว้ล่วงหน้า

ในข้อมูลอินพุตและรวมเข้าด้วยกัน เพื่อป้องกันไม่ให้มีการแบ่งและ การรวมลำดับย่อยที่จัดเรียงไว้ล่วงหน้าเพิ่มเติมโดยไม่จำเป็น องค์ประกอบอินพุตที่เรียงลำดับทั้งหมดจากน้อยไปมากจึงถูกจัดเรียงใน O(n)

## **2.2 ตัวอย่าง Natural Mergesort**

ภาพตัวอย่างต่อไปนี้ แสดง Natural Merge Sort โดยใช้ลำดับ [3, 7, 1, 8, 2, 5, 9, 4, 6] เป็นตัวอย่าง



# **3. Balanced merge**

แม้ว่าดิสก์ไดรฟ์ส่วนใหญ่สามารถทำงานกับไฟล์ชั่วคราวจำนวนมากได้ แต่ก็ใช้ไม่ได้กับพื้นที่จัดเก็บเทปเช่นกันเป็นเรื่องยากที่จะมีไฟล์หลายไฟล์ในเทปเดียว และเรามีเทปไดร์ฟในจำนวนจำกัด เราสามารถทำได้ด้วย 3 หรือ 4 เทป แต่เราสามารถเพิ่มประสิทธิภาพได้ด้วยเทปไดร์ฟมากขึ้น

## **3.1 Two-way merge**

### **3.**1.**1 k-way merge คือ** อัลกอริธึมการ merge k-way หรือการผสานหลายทางเป็นประเภทเฉพาะของอัลกอริทึมการผสานลำดับที่ ใช้ในการ k sorted list และรวมเข้าเป็น single sorted list

อัลกอริทึมการ merge เหล่านี้โดยทั่วไปหมายถึงอัลกอริธึมการผสานที่ใช้รายการเรียงลำดับมากกว่าสองรายการ ซึ่งการผสานแบบสองทางเรียกอีกอย่างว่าการรวมแบบไบนารี่

### **3.1.2 Two-way merge** การผสานแบบ 2 ทางหรือการผสานแบบไบนาร ใช้ในการจัดเรียงการ merge ตามที่ได้กล่าวมาแล้วใน 3.3.1 คือ Classic merge ที่ปรากฎขึ้นบ่อยครั้งในการจัดเรียงการผสาน Classic merge จะแสดง รายการข้อมูลด้วย คีย์ต่ำสุดในแต่ละขั้นตอน ให้เรียงลำดับบางรายการ จะสร้างรายการที่เรียงลำดับที่มีองค์ประกอบทั้งหมดในรายการที่เรียงลำดับที่มีองค์ประกอบทั้งหมดในรายการอินพุตใดๆ และทำำในเวลาตามสัดส่วนกับผลรวมของความยาวของรายการอินพุต

### **3.1.3 Iterative 2-way merge** การผสาน 2 ทางแบบวนซ้ำ เป็นการผสานอาร์เรย์ k สองชุดซ้ำๆ โดยใช้การผสานแบบ 2 ทางจนเหลือเพียงอาร์เรย์เดียว หากอาร์เรย์ถูกรวมในลำดับที่ต้องการ เวลาทำงานที่ได้จะเป็น O(kn) เวลาทำงานสามารถปรับปรุงได้โดยการรวมครั้งแรกกับครั้งที่สอง ครั้งที่สามกับครั้งที่สี่ และอื่นๆ เนื่องจาก จำนวนอาร์เรย์ลดลงครึ่งหนึ่งในการวนซ้ำแต่ละครั้ง ทุกองค์ประกอบจะถูกย้ายเพียงครั้งเดียว เวลาทำงานต่อการวนซ้ำจึงอยู่ใน Θ(n) เนื่องจาก n คือจำนวนขององค์ประกอบ เวลาทำงานทั้งหมดจึงเป็น Θ(n log k)

# **4. Polyphase merge**

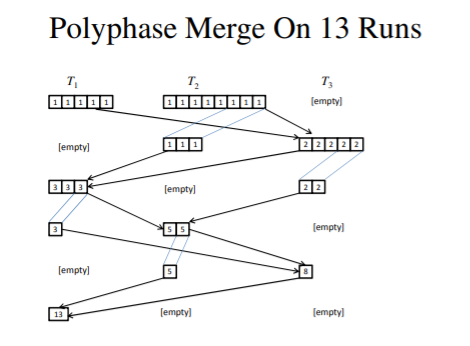
การเรียงลำดับแบบผสานหลายเฟสเป็นรูปแบบของการเรียงลำดับการผสานจากล่างขึ้นบนที่เรียงลำดับรายการโดยใช้การกระจาย รายการย่อย (runs) ที่ไม่สม่ำเสมอ ซึ่งส่วนใหญ่ใช้สำหรับการเรียงลำดับภายนอก และมีประสิทธิภาพมากกว่าการเรียงลำดับผสานทั่วไปเมื่อมีน้อยกว่า 8 ไฟล์การทำำงานภายนอก (เช่น เทปไดร์ฟหรือไฟล์บนฮาร์ดไดร์ฟ) การเรียงลำดับการรวมหลายเฟสไม่ใช่การเรียงลำดับที่เสถียร

## **4.1 Polyphase merge example**

สมมุติว่าเรามี 3 เทป (T1, T2, T3 ) และเราจะ merge ตามลำดับดังนี้

1. จัดเรียงและแจกจ่ายบันทึกไปยัง T1 และ T2
2. Merge T1 และ T2 ไปยัง T3 เหลือไว้บางตัวใน T2
3. Merge T2 และ T3 ไปยัง T1 เหลือไว้บางตัวใน T2
4. Merge T3 และ T1 ไปยัง T2 เหลือไว้บางตัวใน T3
5. Merge T1 และ T2 ไปยัง T3 เหลือไว้บางตัวใน T2 และอื่นๆ

เรามักทิ้งเทปต้นฉบับไว้สองอันและเทปหนึ่งอันสำหรับวางไฟล์ที่ผสาน



# **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 กันยากัน 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 กันยากัน 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). [ชีทสรุป 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>