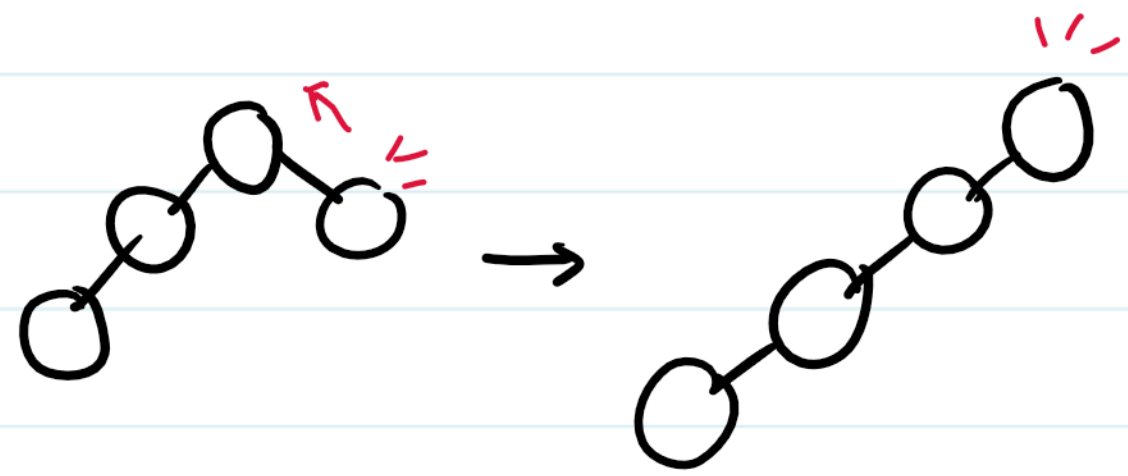


Data 21/9/2021

Splay tree (10/10)

Insert with self-adjustment feather

→ $O(n)$, $O(n)$



เป็น Degenerate tree ได้ เลย Balance 100

Analysis

- มง Balance มั่วๆ ใช้ $O(M \log N)$ เมื่อ $M \geq N$
- คล้าย $O(\log N)$
- Node ที่ใช้จุดเชื่อมต่อคือค่าๆ กันจะมาอยู่ด้วยกัน

Compare

- run time คล้ายคล้ายกับ AVL
- Implement ง่ายกว่า
- ไม่ต้องทำเรื่อง balance

ข้อเสีย

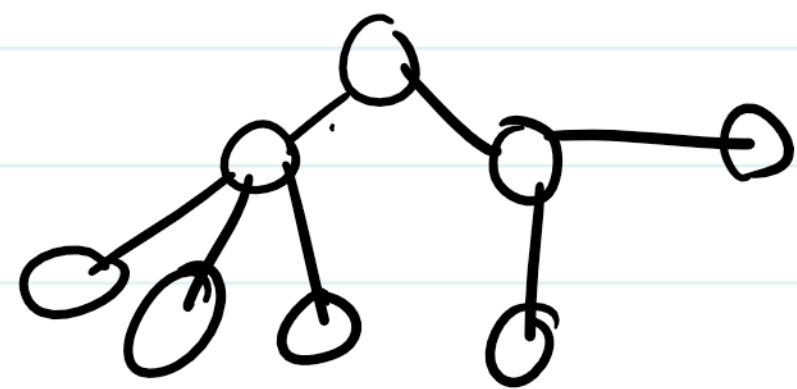
- ไม่สามารถเปลี่ยน tree แล้ว

B Tree

- ใช้ implement ข้อมูล Hard disk
- เก็บค่า Balance อยู่แล้ว (find, insert, delete → logarithm time)
- 1 node มีลูกมากกว่า 2 ก็ได้ คือได้ ~

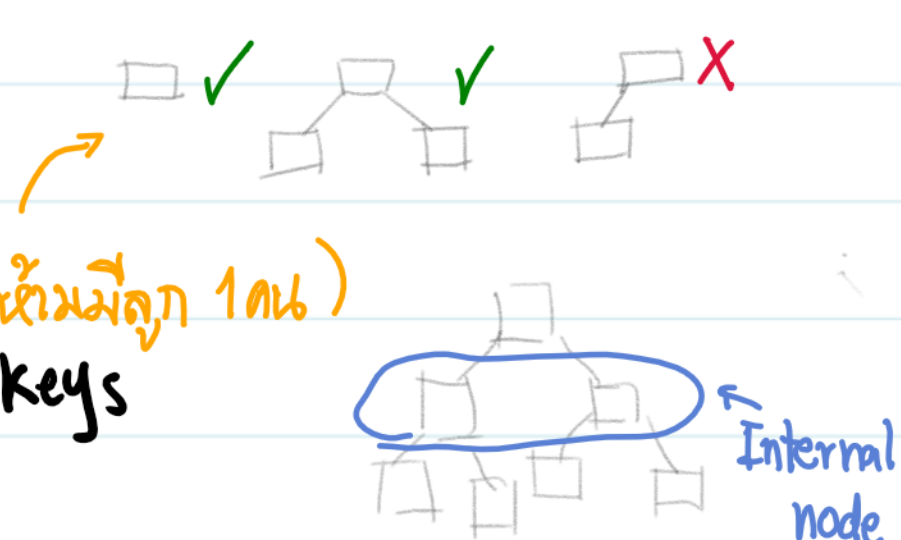
M-ary Tree

$M = \text{order}(\text{ลูก})$



B Tree with order M

1. ทุก node มีลูกมากกว่า 1 คือ M ตัว
2. node ที่มี k key จะมีลูกทั้งหมด $k+1$
3. root node มีลูกอย่างน้อย 2 ลูก ยกเว้นลูกเป็นใบ (ถ้าเป็นคือ จะมีมีลูก 1 คน)
4. Internal node มีลูกอย่างน้อย $\lceil M/2 \rceil$ ตัว และ $\lceil M/2 \rceil - 1$ keys
5. ทุกๆ ใบจะเกิดที่ level เดียวกัน
6. (Optional) ทุกๆ ใบจะต่อมี $\lceil M/2 \rceil - 1$ keys



* $\min \lceil M/2 \rceil - 1$ keys
max $M - 1$ keys

B-tree Insertion

1. เริ่มที่ root เหมือน BST, ไปหาช่องว่างใส่ใบเสมอ
2. Overflow?
 - no, ใส่ได้เลย!!
 - yes, ยัดไปก่อน → ย้ายมาตรงกลาง → split 2 ตัว เป็น parent

Example constion of B-Tree ออกสอบ!!

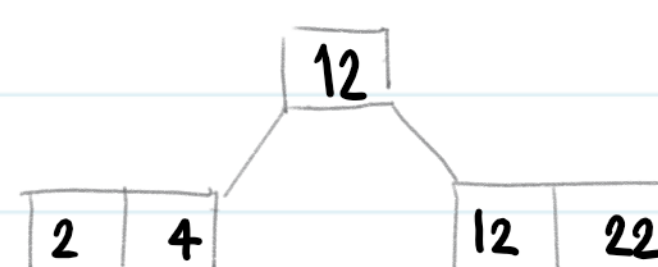
B-Tree of order 5

2 4 12 14

Insert (22):

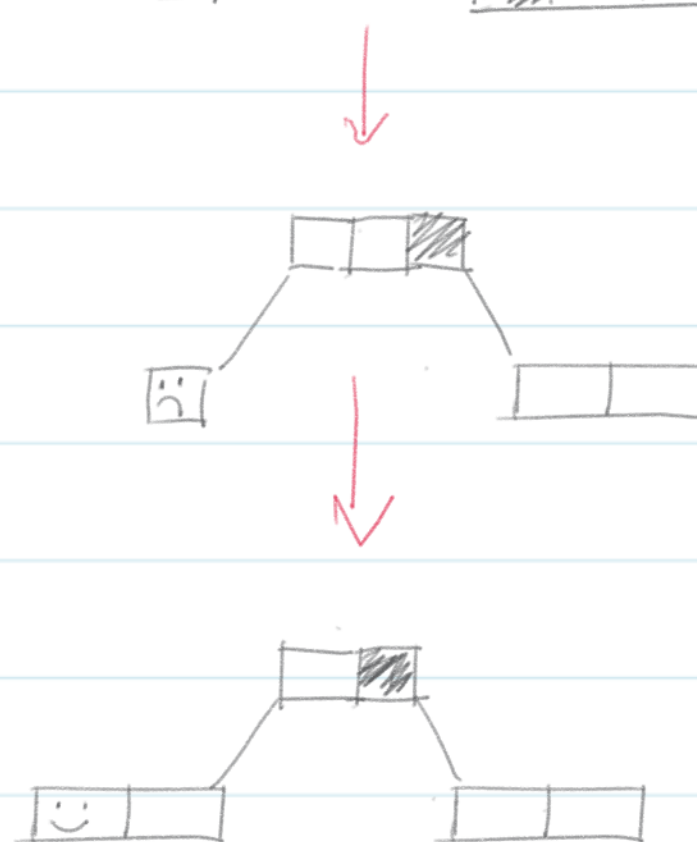
2 4 12 14 22

เก็บแล้วหามาตรงกลาง

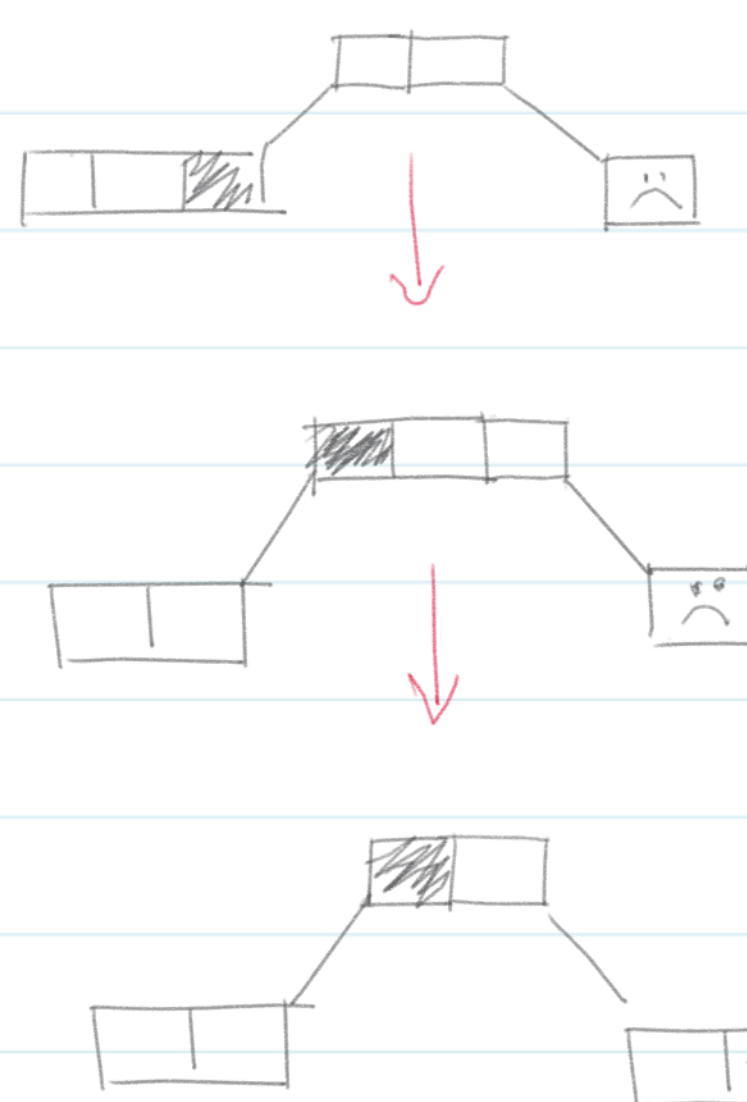


B-Tree Deletion

1. เริ่มจาก root
2. ถ้า node นั้นเป็นใบลบเลย
3. ถ้า Internal node อยู่เพื่องลบ
 - หา Min. RST มาแทน แล้วลบ
4. ถ้า Underflow
 - ย้าย node เข้าที่ว่างมาแทนว่าจริงได้



- ถ้าไม่ได้ ให้ยืมช่วย



- ถ้ายังไม่ได้ ให้ดึง parent ลงมาแล้วลบรวมเลย !!