Priority Queue

Processing the shortest job first

⇒ extracting min

Also. insert new printing job into Queue

∴ Priority Queue { Extract-Min / Insert.

⇒ Insert → [Priority Queue] → Extract-Min

Possible Implementation

1. Unsorted
   - Insert : O(1) (just put it at the end!)
   - Extract : O(n). (traverse).
2. Sorted
   - Insert : O(n) (ascending order)
   - Extract : O(1). first element!

⇒ Is there any data structure

⇒ Insert, Extract both O(logn)?

(Binary) Heap.

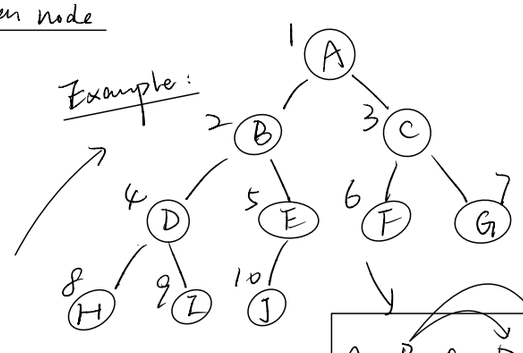
- Min-heap : Parent node ≤ children node
- Max-heap

root in array position 1

For any element in position i:

- left children node: 2i
- right children node: 2i+1
- parent node ⌊i/2⌋

Example:



We can have one-to-one mapping from an array to a heap!

→ Insertion

① Add new element to the next available position at the lowest level

② "Bubble Up":

if parent of the element > element then: (min-heap)

interchange parent with children (swap)

(log n) levels in total. (n elements)

⇒ running time: Θ(log n)

→ Extract-Min:

1. Copy the last element X to the root

   ⇒ overpwe root!
2. "Bubble down": "가라앉아"

   if larger than either of its children:

   swap with the smaller one (why smaller one? if large one.

   after swapping "larger / smaller". doesn't satisfy min-heap's rule!"

⇒ running time: Θ(log n)

Heap Sort : ① perform n times insert.

② perform n times Extract-Min

⇒ Θ(nlogn)