Lecture 18 - Nov 17

Graphs

Priority Queues ADT: Introduction
Heap: Structural Property
Heap: Relational Property

Announcements/Reminders

- Today's class: notes template posted
- **Assignment 2** released
- Change of Dates:
  - + **Assignment 2** to be due on Wed, Nov 19
  - + **Test 2** to be take place on Mon, Nov 24
- Online Course Evaluation

Wednesday class: ≈ 40-50 min lecture.
≈ ≤ 20 min. Q & A ~ or office Zoom hour.

Test 2 (WSC 106, 4:30 PM to 5:20 PM, Monday Nov 24)

- **Coverage**
  - + Graphs lecture (slides 33 – 72, notes, example code)
  - + Tutorials Weeks 9 and 10
  - + Assignment 2
- **Format**
  - + **Programming** Part (Eclipse):
    - * Import a Java starter project (like A2)
    - * Implement Java classes/methods to pass test cases
    - * e.g., Implement graph op from scratch.
    - * e.g., Implement graph op based on given DFS (A1) or BFS (A2).
  - + **Written** Part (eClass):
    - * MCQs ✓
    - * Written questions ✓ (e.g., short answers, justifications, proofs)

lecture on PQ & heap.

incident edges

ref. solutions

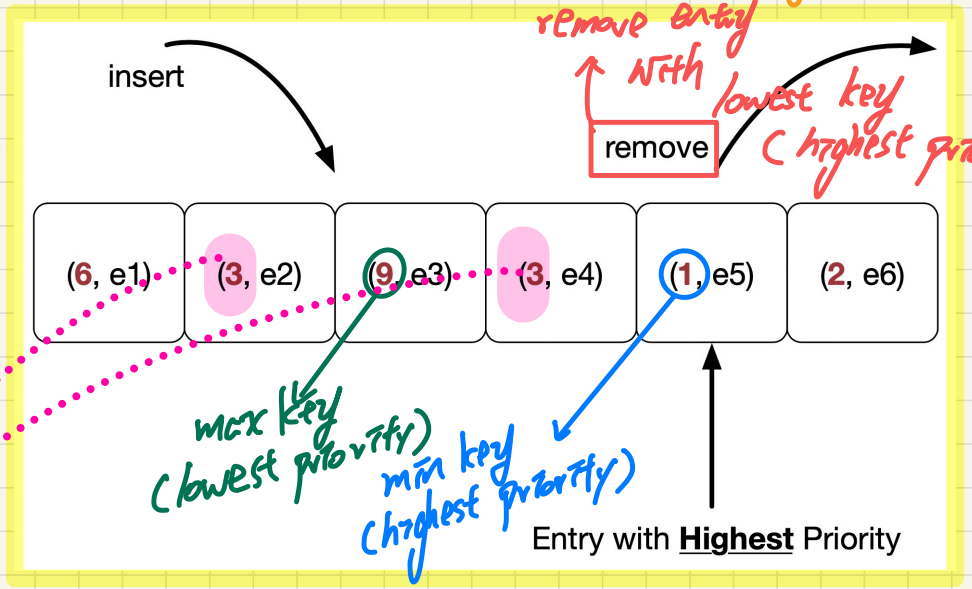
HashMap HashSet ArrayList (API page accessible)

What is a Priority Queue (PQ)

min-key PQ.

key ↓ priority ↑

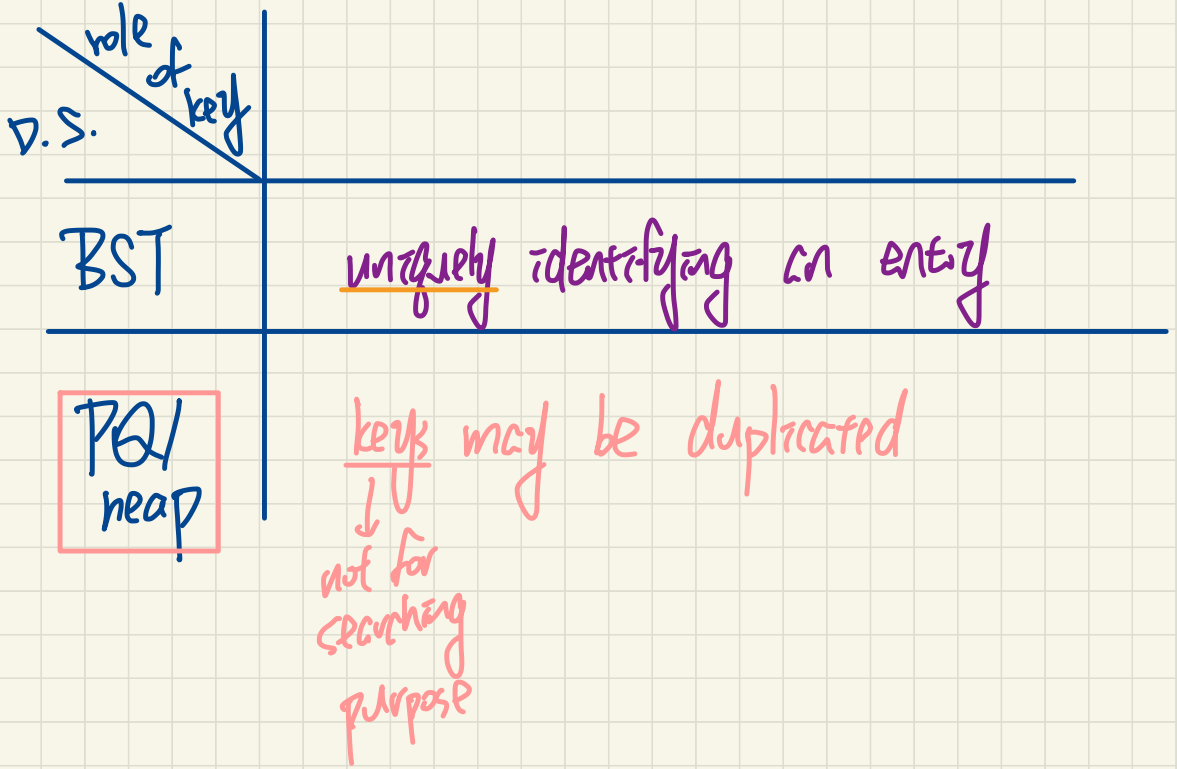
e.g. D(v) → min D(v) chosen first.



entries with the same priority (does not matter which one to choose)

Compare PQ with FIFO Queue

1. entries in PQ removed according to priority values
2. entries in FIFO queue removed according to chronological order of insertions.


role of key
D.S.
BST
uniquely identifying an entry
PQ/ heap
keys may be duplicated
not for searching purpose

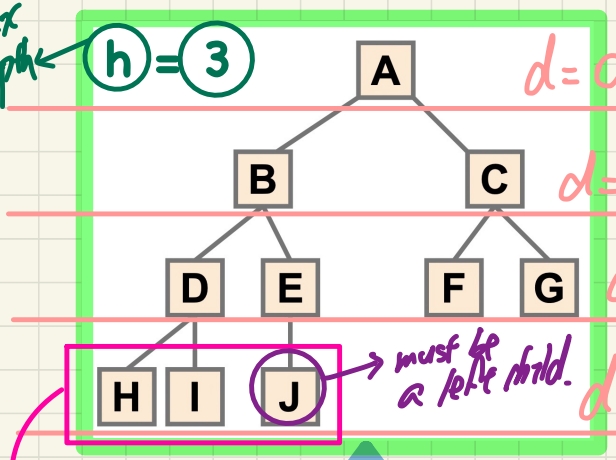
$*(2^0 + 2^1 + \cdots + 2^{h-1}) = 2^h - 1$

BT Terminology: Complete vs. Full BTs

C1: Level 0 ~ Level h-2
↳ each node has 2 child nodes

Complete BT
where at level h, we have 2^h nodes

terms: $S_k = 2^k - 1$



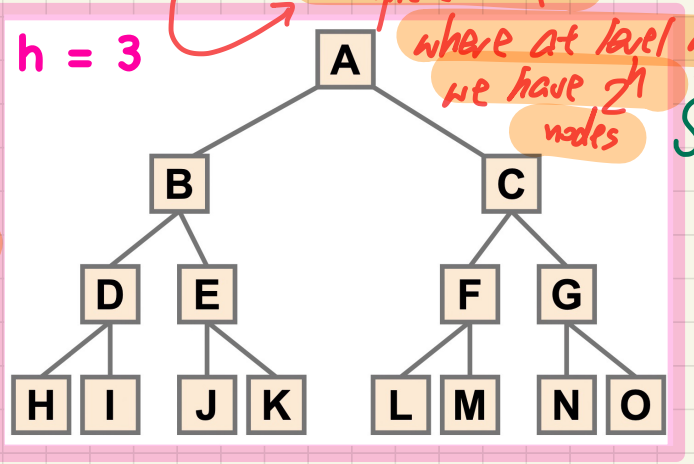
level

h-3

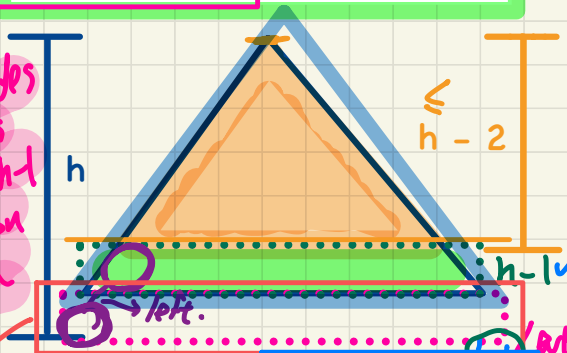
h-2

h-1

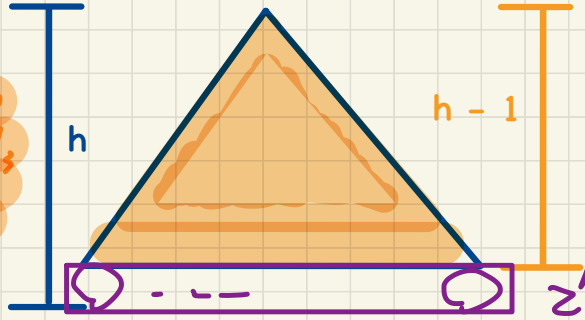
h



C3: child nodes of nodes at level h-1 filled Lon ↳ to R



C2: each node has 0, 1, 2 child nodes



max: 2^h

Min # nodes? $(2^0 + 2^1 + \cdots + 2^{h-1}) + 1 = 2^h$

Max # nodes? $(2^0 + 2^1 + \cdots + 2^h) + 2^h = 2^{h+1} - 1$

$2^h - 1$

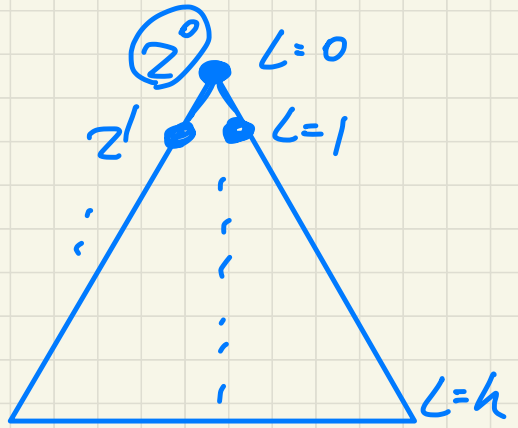
Min # nodes? $(2^0 + 2^1 + \cdots + 2^h) = 2^{h+1} - 1$

Max # nodes?

Geometric Sequence

$$S_k = \frac{1 \cdot (r^k - 1)}{r - 1}$$

- 1: first term
- r: common factor
- k: # terms



In case of BT with height = h

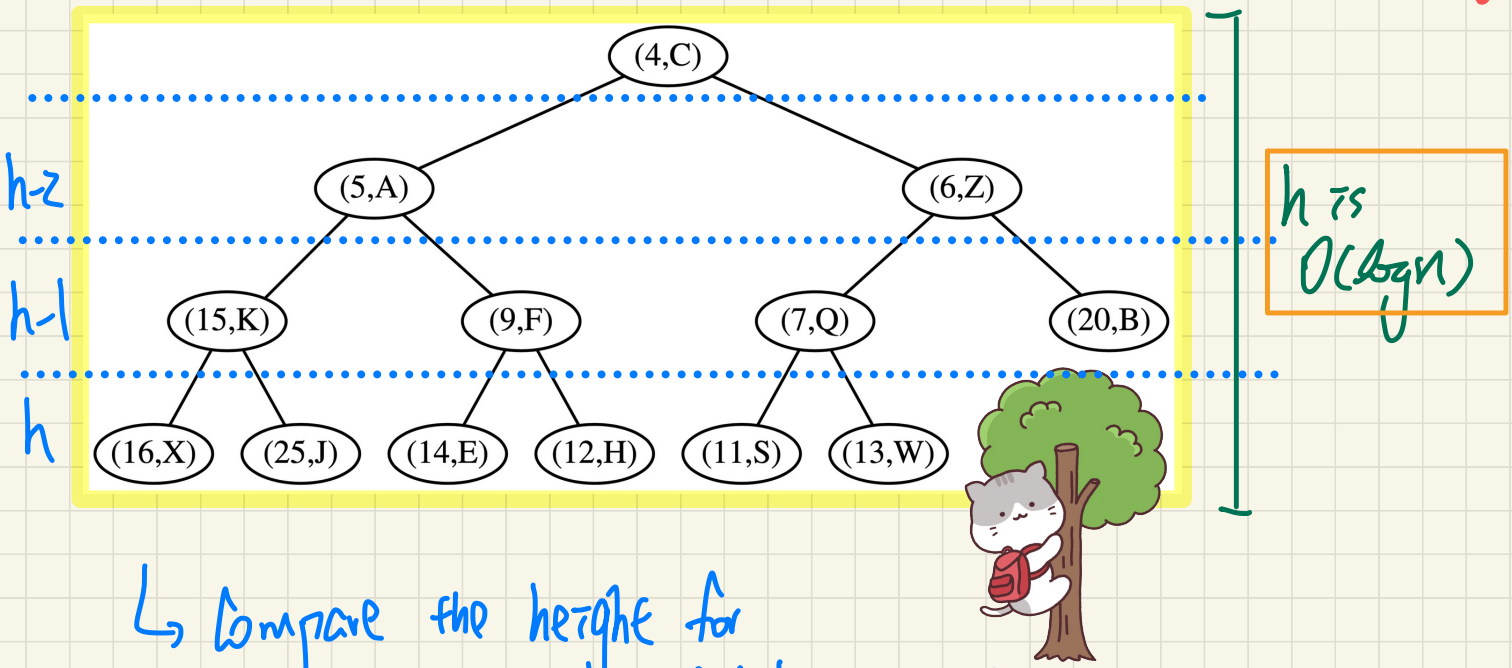
$$\frac{1 \cdot (2^{h+1} - 1)}{2 - 1} = 2^{h+1} - 1$$

Heaps: Structural Properties of Nodes

$2^{h+1} - 1 = n$ max # nodes

Property: The tree is a complete Binary Tree

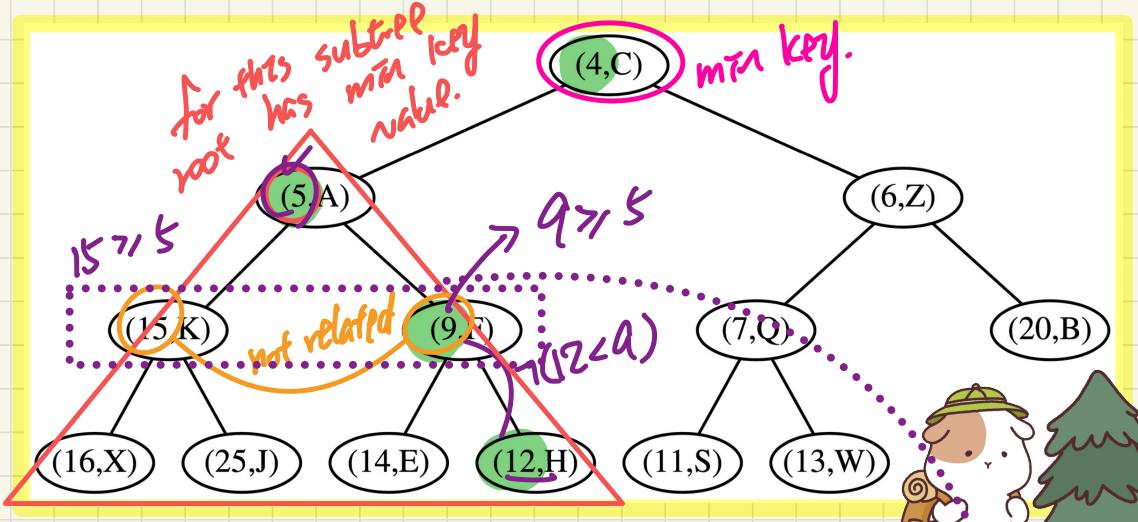
↳ h is $O(\log n)$



↳ Compare the height for self-balancing BSTs.

Heaps: Relational Properties of Keys

Property: Each non-root node n is s.t. key(n) ≥ key(parent(n))



P1 Any leaf-to-root path generates a sorted seq. of keys. (non ascending order)

↳ child key = parent key
child key > parent key

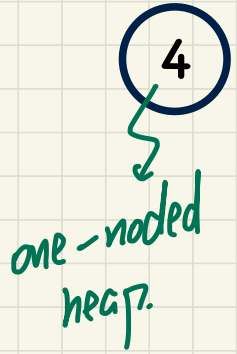
P2 min key stored in the root.

P3 keys between LST and RST are not related

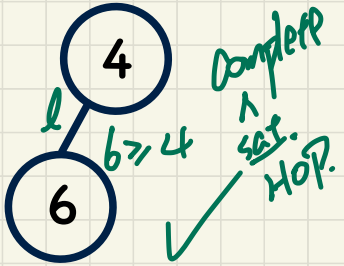
↳ the only property is w.r.t. the parent

Example Heaps

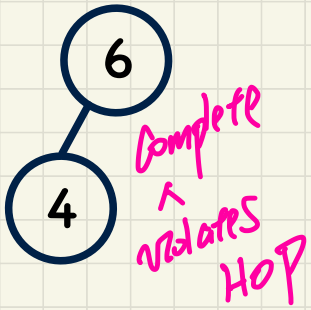
Example 1



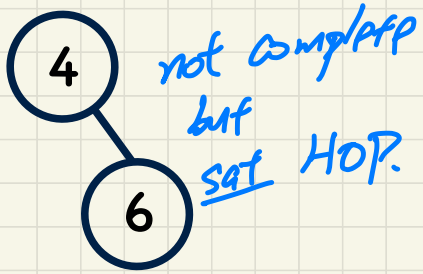
Example 2



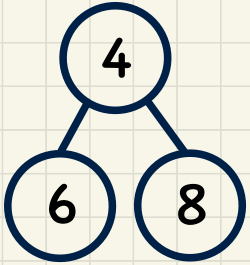
Example 3



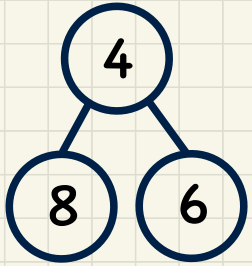
Example 4



Example 5



Example 6



heaps!