COL 106 Lecture 9

Topic: Priority Queues

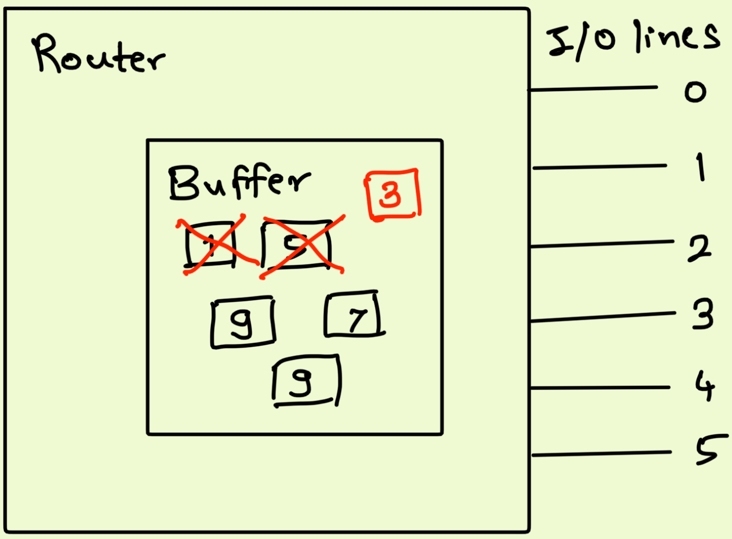
Attendance Marking Protocol

CS – 11:00 – 11:20

EE – 11:20 – 11:40

MT – 11:40 – 12:00

Motivating Example



- Receive incoming packets in buffer.
- Send packets one-by-one.
  - most important packet first!

Recap: Queues

- Enqueue
- Dequeue

  (First-in-first-out)

- isEmpty, len, etc

Want: Priority Queues

- Enqueue
- Extract Min

  (Remove the minimum)

- isEmpty, len, etc.

Universe of items must be an ordered-set

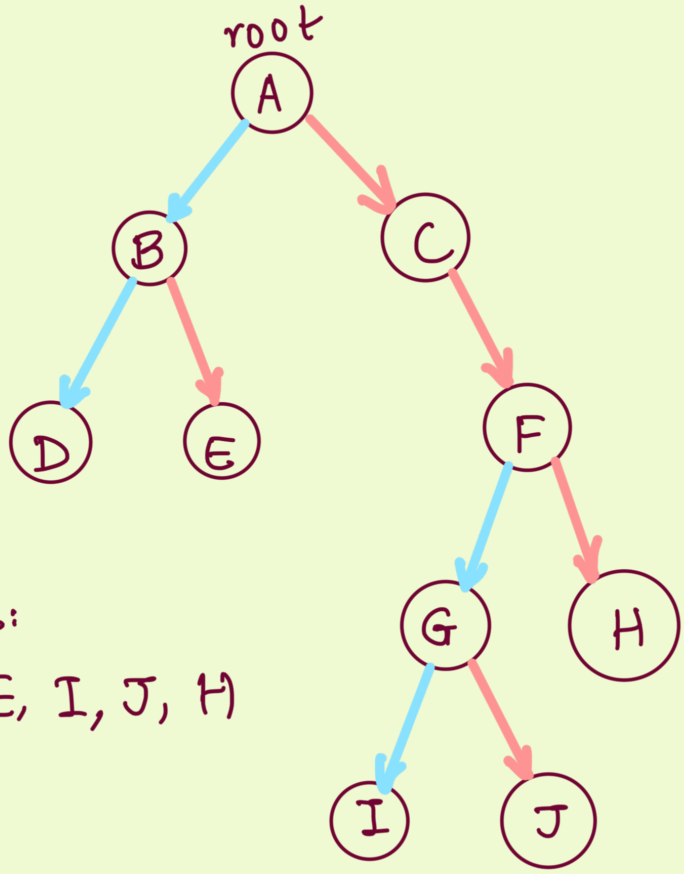
$(S, \leq)$

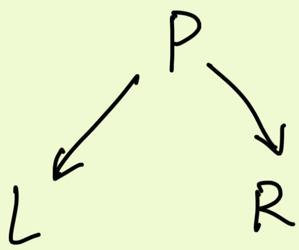
For all $x, y, z \in S$ if $(x \leq y)$ and $(y \leq z)$ then $(x \leq z)$

Implementations?

	Enqueue	Extract Min
Unsorted List / Array, Linked List	$O(1)$	$O(n)$
Sorted List / Array / Linked list.	$O(n)$	$O(1)$

Binary Trees





P: parent of L, R

L: left child of P

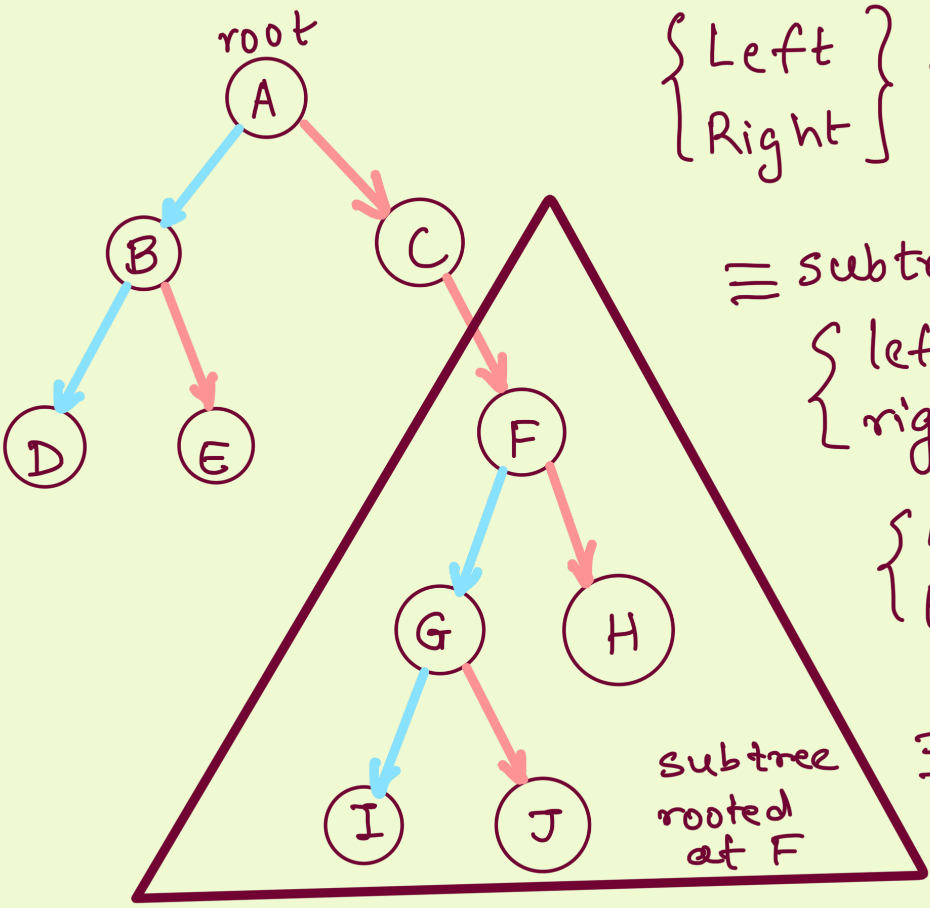
R: right child of P.

leaves:
D, E, I, J, H

Ancestors of C:
A, C

Descendants of C
C, F, G, H, I, J.

Subtrees



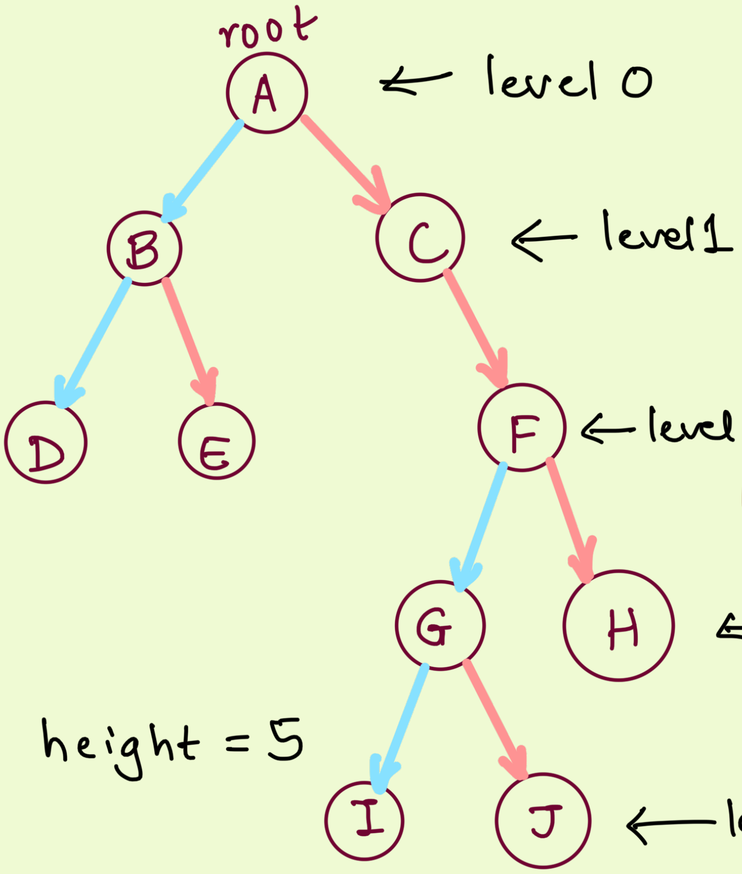
$\left\{\begin{array}{l}\text{Left}\\ \text{Right}\end{array}\right\}$ subtree of X

$\equiv$ subtree rooted at $\left\{\begin{array}{l}\text{left}\\ \text{right}\end{array}\right\}$ child of X.

$\left\{\begin{array}{l}\text{Left}\\ \text{Right}\end{array}\right\}$ descendant of X

$\equiv$ node in the $\left\{\begin{array}{l}\text{left}\\ \text{right}\end{array}\right\}$ subtree of X.

Levels, Depth, Height



Depth of X : distance between root and X

Height of a tree : # nodes on a longest root-to-leaf path.

$$\text{Height} = 1 + \max_{X:\text{ node}} \text{depth}(X)$$

$$= 1 + \#\text{ levels}$$

Claim : Let n be the # nodes and h be the height of a binary tree. Then

$$\log_2(n+1) \le h \le n \le 2^h - 1$$

Proof :

if $\ell \le h-1$

$$1 \le \#\text{ nodes in level } \ell \le 2^\ell$$

Sum up over $\ell = 0, \cdots, h-1$.

$$h \le n \le 2^h - 1$$

Implementation of Binary Trees

class BinaryTreeNode:

Attributes: key, $\underbrace{\text{left, right, parent}}_{\text{refs to BinaryTreeNode objects.}}$

class BinaryTree:

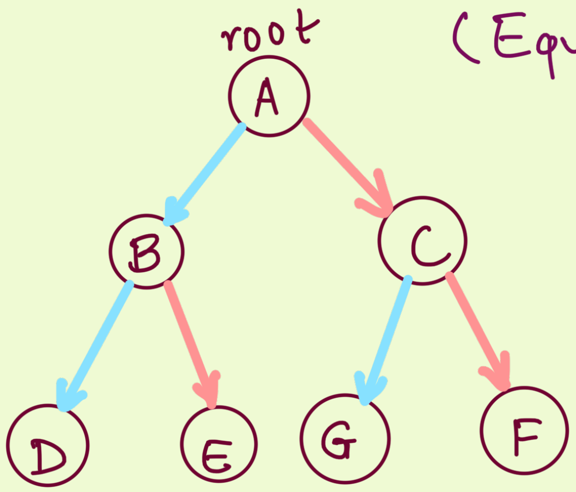
Attribute: root → ref. to a BinaryTreeNode obj.

Complete Binary Tree

Complete binary tree of height h:

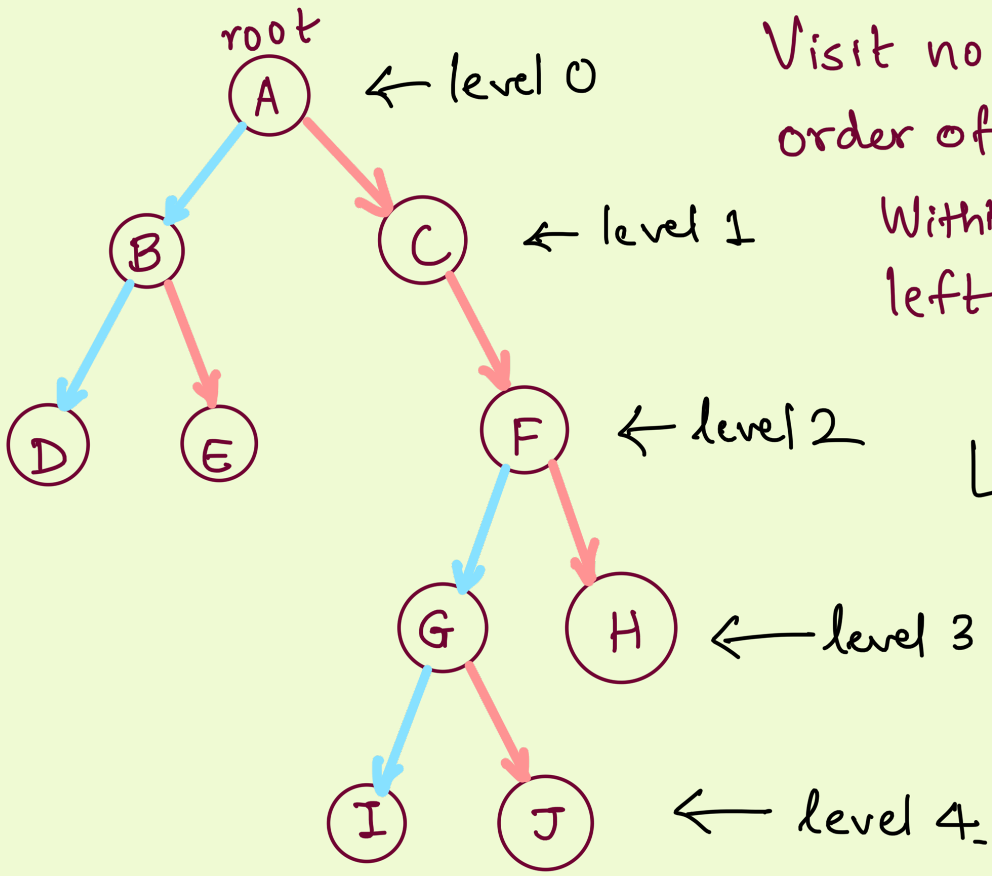
- All nodes in levels 0 ..... h-2 have two children each.
- All nodes in level h-1 are leaves.

(Equivalently: tree with h full levels.)



nodes in a complete binary tree of height h $= 2^h - 1$.

Level-Order Traversal



Visit nodes in increasing order of levels

Within each level from left to right.

Level-order traversal:

A, B, C, D, E, F, G, H, I, J

Level-Order Traversal Algorithm

If root is not None:

  Enqueue root into an empty queue.

  While queue not empty:

    Dequeue a node, say x.

    Visit x.

    If x.left is not None, enqueue x.left.

    If x.right is not None, enqueue x.right.

Exercise: Prove that the above algorithm performs level-order traversal.