KEY POINTS

A heap is a binary tree which stores keys at its nodes:

- Heap order
- Complete binary tree

Inserting to a heap:

For a key k:

- Add a new node, and store k here. This is the last node
- Restore the heap order property, via upheap.

Remove Min

- Set root node to last.
- Downheap the tree.
- Equivalent to removeMin from PQ ADT
- Runs in O(log n) time as depth is O(log n).

NAME/DATE/SUBJECT

Heap Intro

NOTES

Heap Order:

For every internal node v other than the root,

$key(v) \geq key(parent(v))$

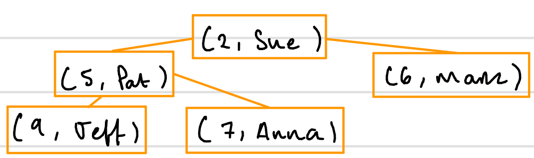
The last node of a heap is the rightmost node of maximum depth.

Complete Binary Tree

Let the height = h;

- for i = 0 ... h-1, there are 2^i nodes of depth i.
- At depth h-1, the internal nodes are to the left of the ext nodes.

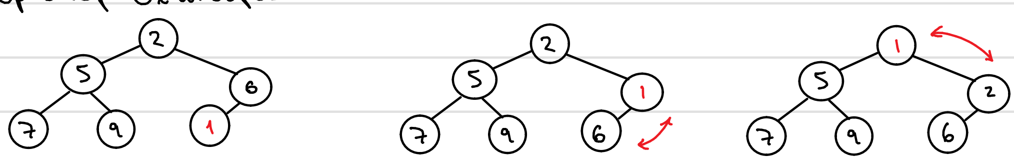
Heap to represent priority queue: store key-value pairs at each node.



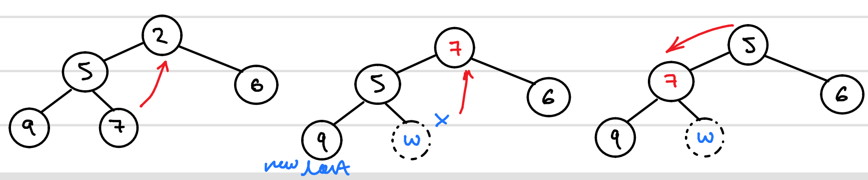
Upheap (O(log n))

- Restores heap-order by swapping a node with its parent until $k_{parent} \leq k$.

Upheap example:



Removing the minimal key: • Set the root = last key, and then remove this last key. You can then downheap.



O(log n)

SUMMARY