Microsoft PowerPoint - [PowerPoint Slide Show - [B tree]]

PQ with Unordered List

- Unordered list of priority values.
- Insertion happens at rear. Thus, O(1)
- To find minimum, search whole list i.e. O(n)
- To delete minimum, search the list first and then delete that element i.e. O(n)

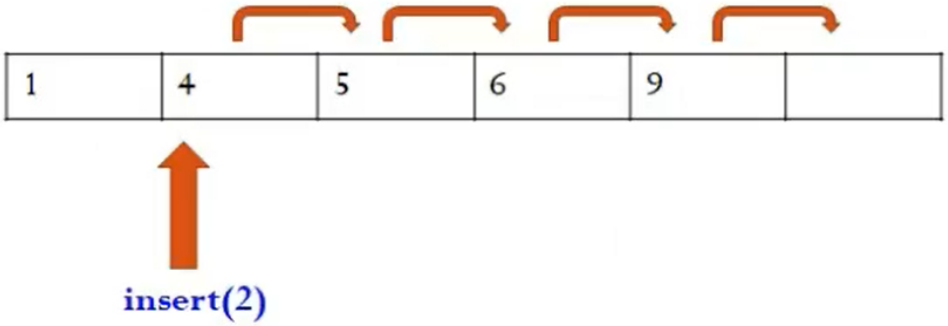
5	4	6	9	1	

insert(1)

Alka Jindal

PQ with Ordered List

- Minimum and delete minimum can be performed in O(1) time.
- However, insertion will take O(n) time



Microsoft PowerPoint - [PowerPoint Slide Show - [B tree]]

(Binary) Heap

- A **complete** binary tree where every node is greater than or equal to its respective parent.
- Because it is complete binary tree, all leaves at last level are filled from left most side.
- The minimum element is at root
- Binary heap is also called min-heap(minimum element at root) .
- Binary heap with maximum element at root is called max-heap.
- Efficiently represented using arrays.

Alka Jindal

Height of Heap

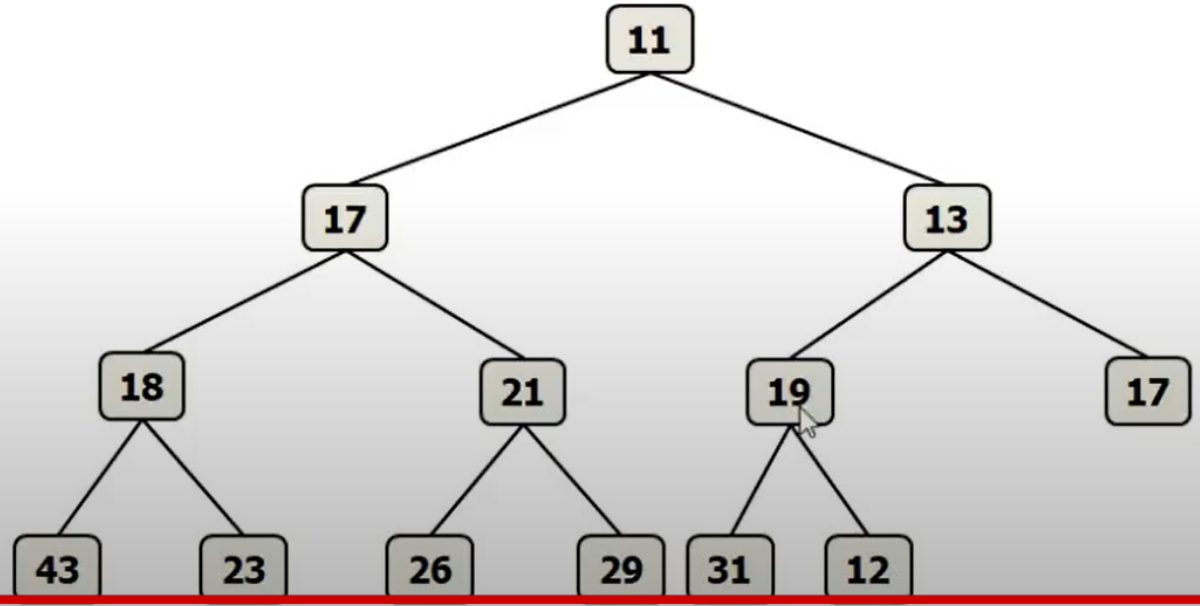
- Suppose heap with **n** nodes has height **h**.
- A perfect binary tree with height h has $2^{h+1} - 1$ nodes and height h-1 has $2^h - 1$ nodes.
- As heap is complete binary tree, therefore

$$2^h - 1 < n \leq 2^{h+1} - 1$$

- $h = \text{floor}(\log_2 n)$

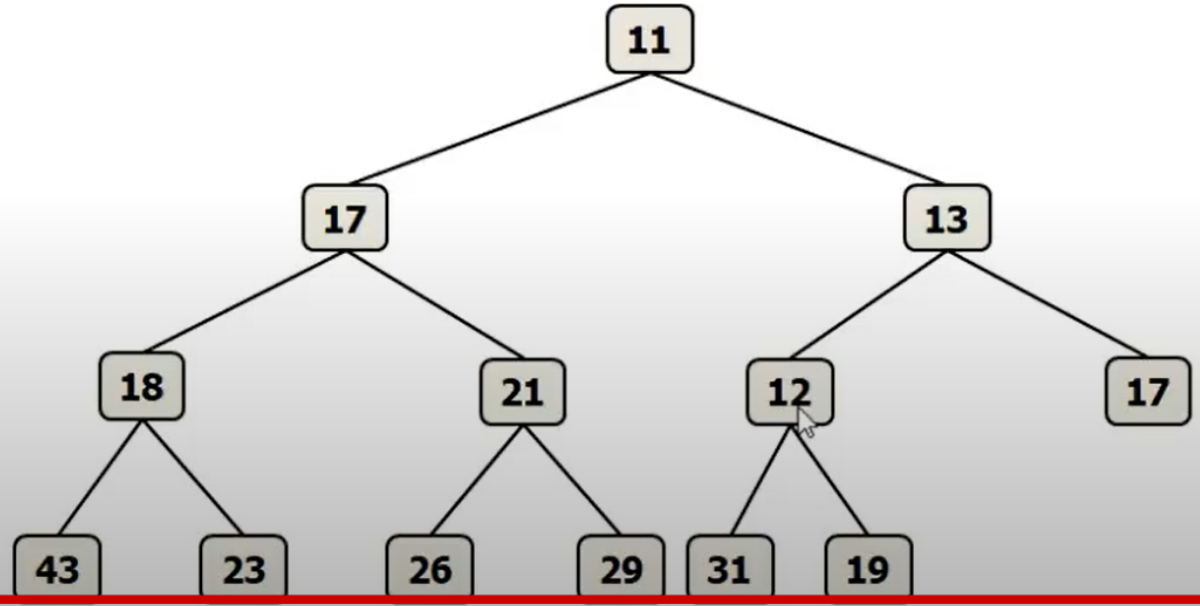
Insertion in Heap : Example

- Insert 12
- Compare 12 with its parent 19. As 19>12, swap values



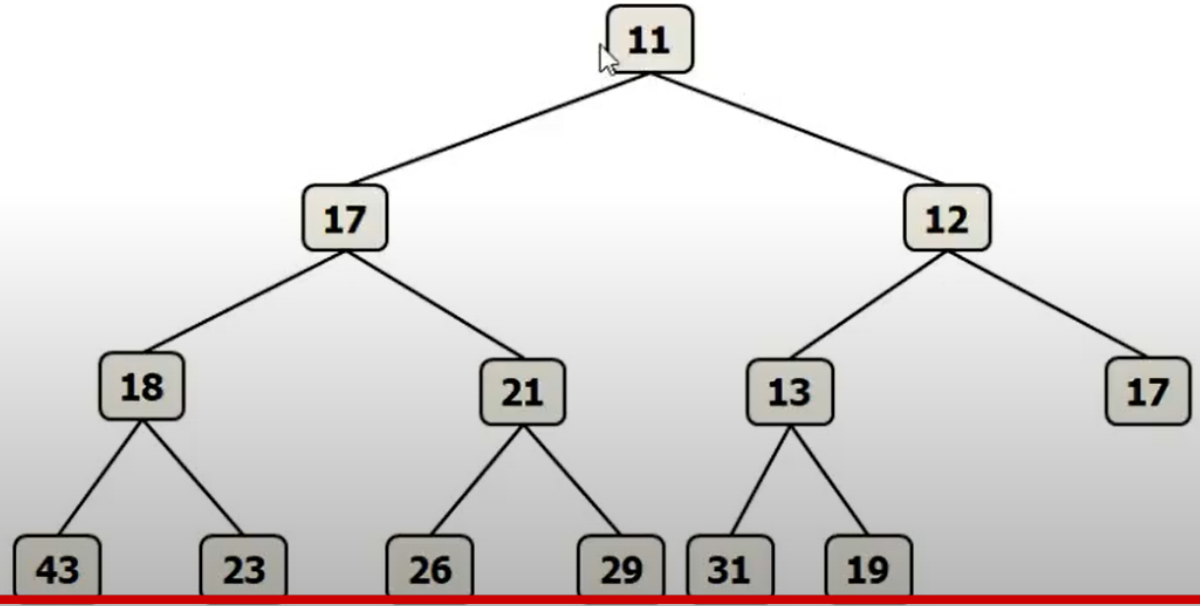
Insertion in Heap : Example

- Insert 12
- Compare 12 with its parent 19. As 19>12, swap values



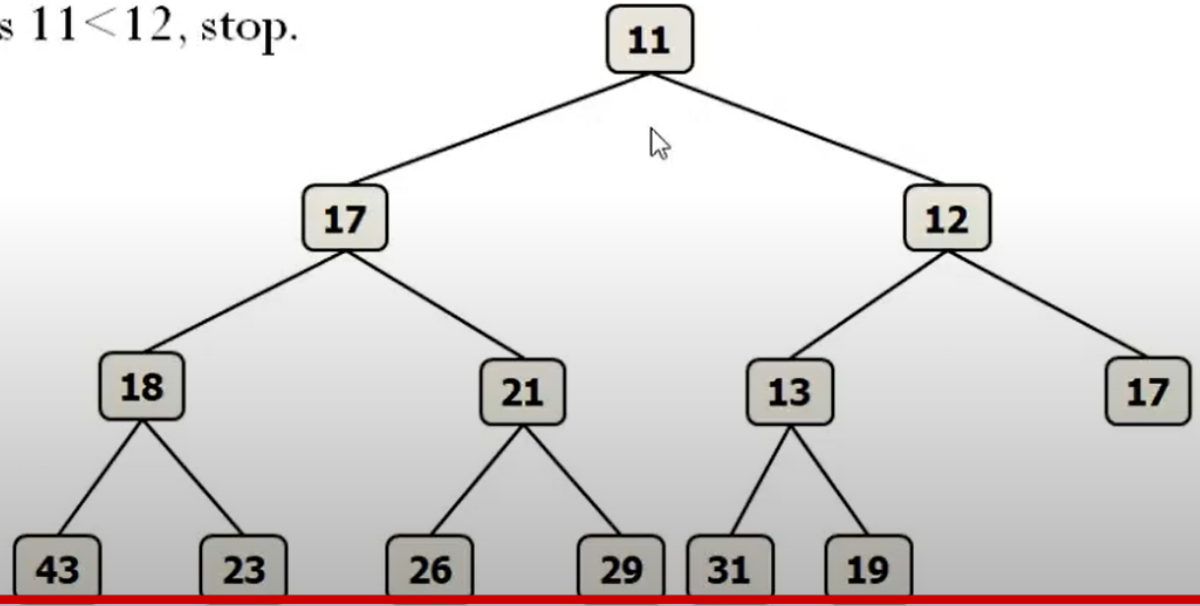
Insertion in Heap : Example

- Insert 12
- Compare 12 with its parent 19. As 19>12, swap values
- Compare 12 with its parent 13. As 13>12, swap values



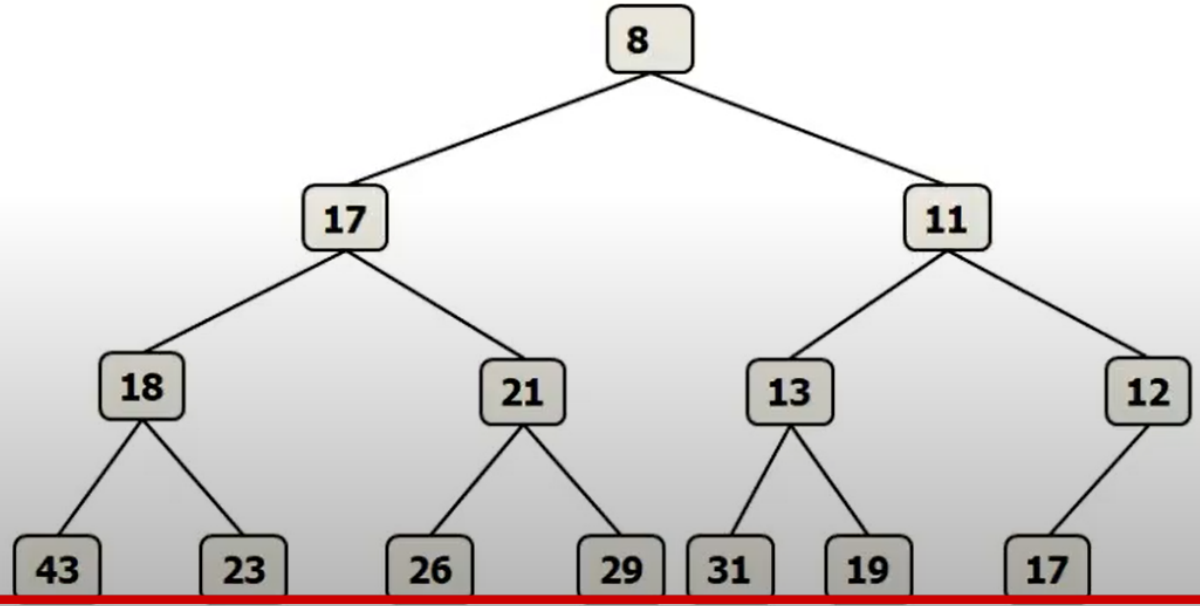
Insertion in Heap : Example

- Insert 12
- Compare 12 with its parent 19. As 19>12, swap values
- Compare 12 with its parent 13. As 13>12, swap values
- As 11<12, stop.



Insertion in Heap : Example

- insert 12
- insert 8



Insertion in Heap : Algo

1. Insert node at last level at leftmost available position.
2. Subsequently compare newly inserted node with its parent. If parent is greater than new node, swap two values.
3. Repeat step 2 for the parent upto root.

Microsoft PowerPoint - [PowerPoint Slide Show - [B tree]]

Insertion in Heap : Analysis

- Adding a node at right place: $O(1)$
- Swapping two values: $O(1)$
- Number of swap operation: $O(\log_2 n)$ (height of heap)
- Total time complexity = $O(\log_2 n)$

Alka Jindal