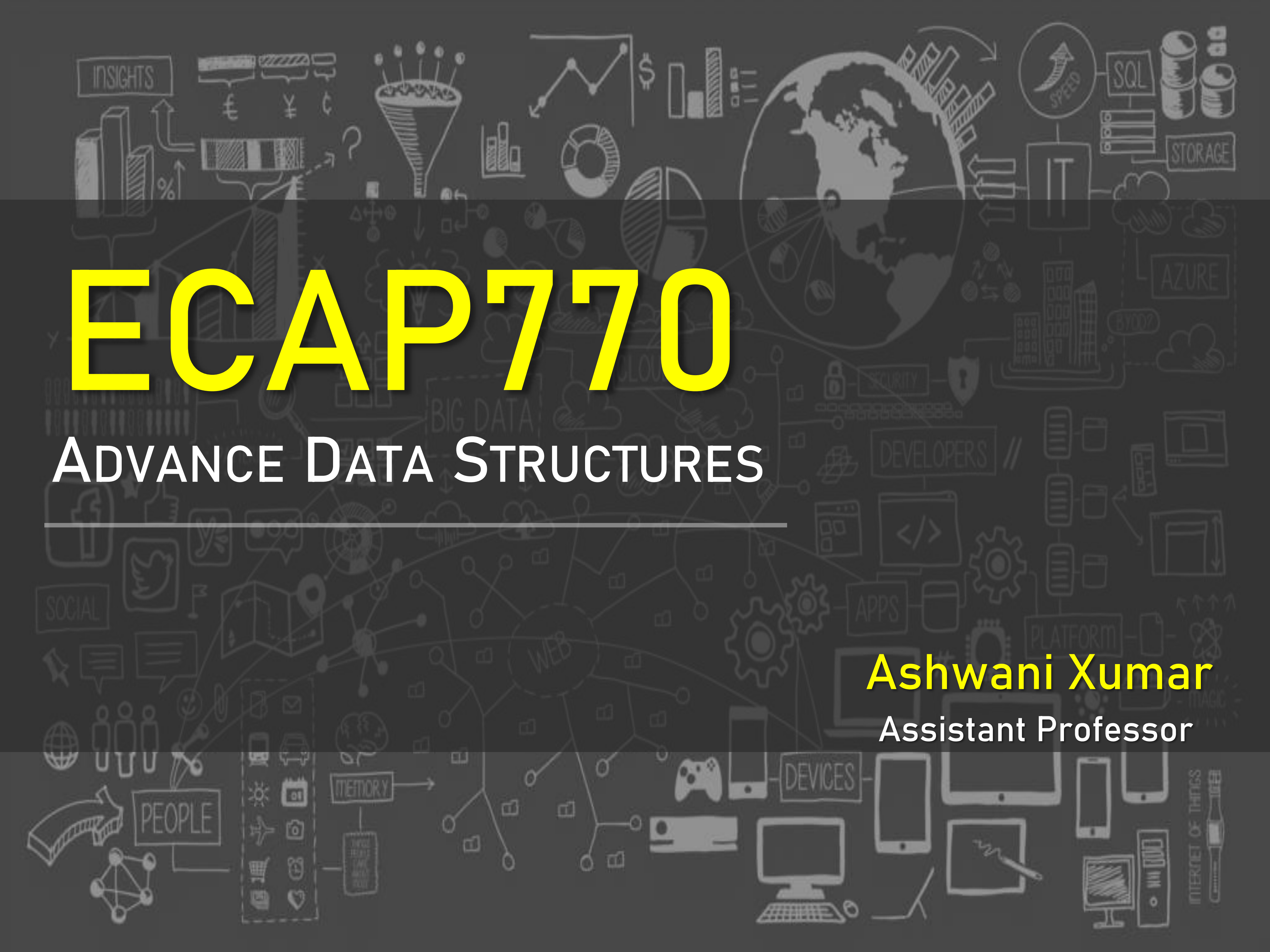


ECAP770

Advance Data Structures

Ashwani Kumar

Assistant Professor

Learning Outcomes

After this lecture, you will be able to

- Understand Heap Data Structure

Heap

- Heap data structure is a complete binary tree that satisfies the heap property.
- Structural property
- Ordering property
- In a complete binary tree, all levels are full except the last level, i.e., nodes in all levels except the last level will have two children and all the nodes should be left-justified.

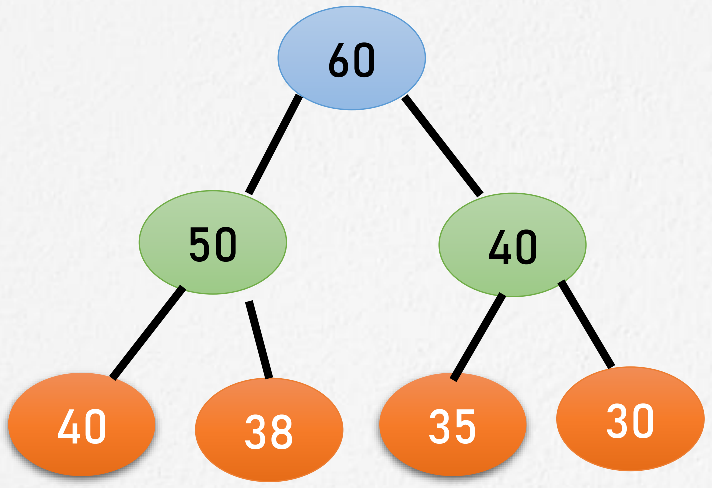
Types of Heap

- Max-Heap
- Min-Heap

Max Heap

- The key present at the root node must be greatest or equal to the keys present at all of it’s children.
- The same property must be true for all sub-trees in that Binary Tree.

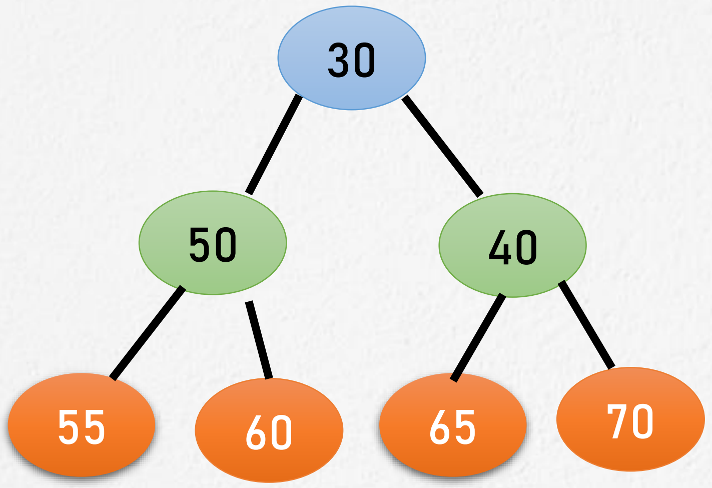
Max-Heap



Min-Heap

- The key present at the root node must be minimum or equal to the keys present at all of it’s children.
- The same property must be true for all sub-trees in that Binary Tree.

Min-Heap



Heap Tree construction

1. Create a new node at the end of heap.

2. Assign new value to the node.

3. Compare the value of this child node with its parent.

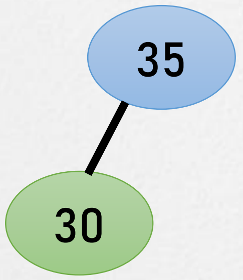
4.If value of parent is less than child, then swap them.

5. Repeat step 3 & 4 until Heap property holds.

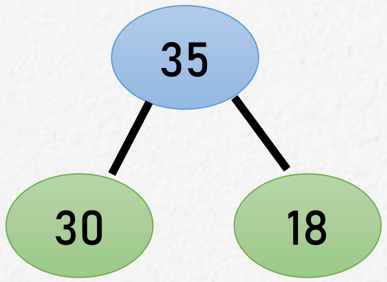
Heap Tree construction (Max Heap)

Data: 35 30 18 26 40 28 10

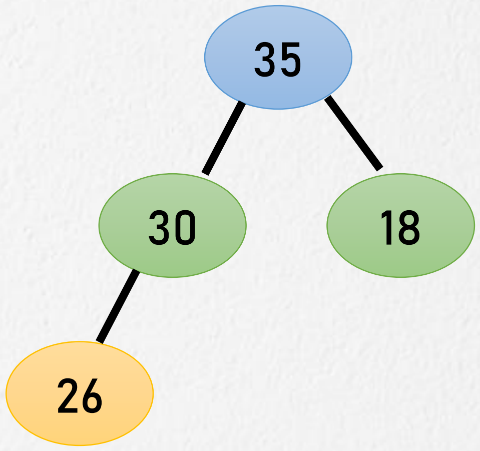
Complexity: O(n log n)



Step 1



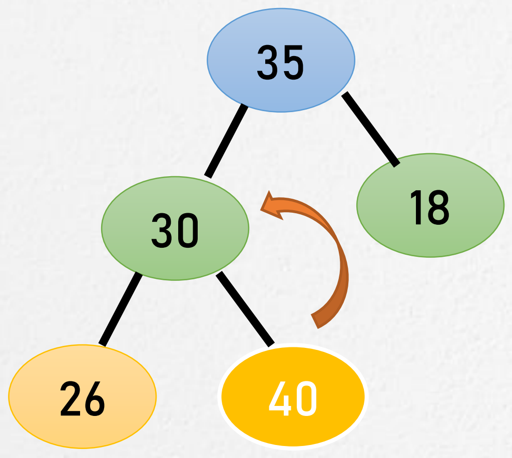
Step 2



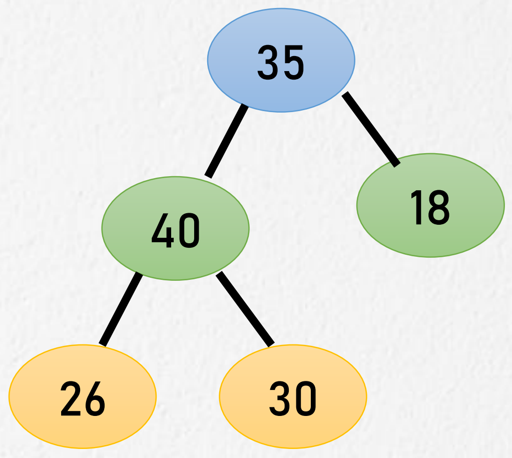
Step 3

Heap Tree construction

Data: 35 30 18 26 40 28 10



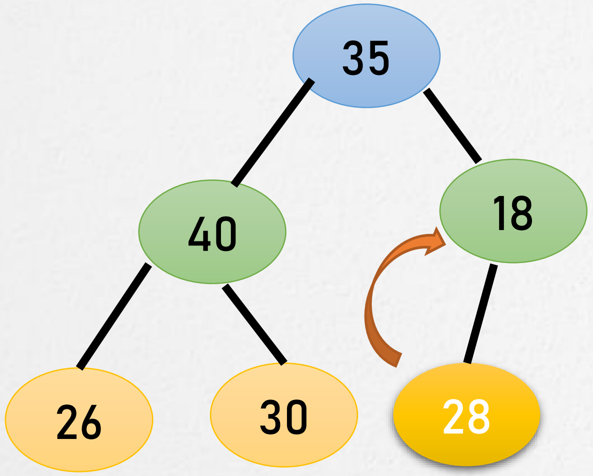
Step 4

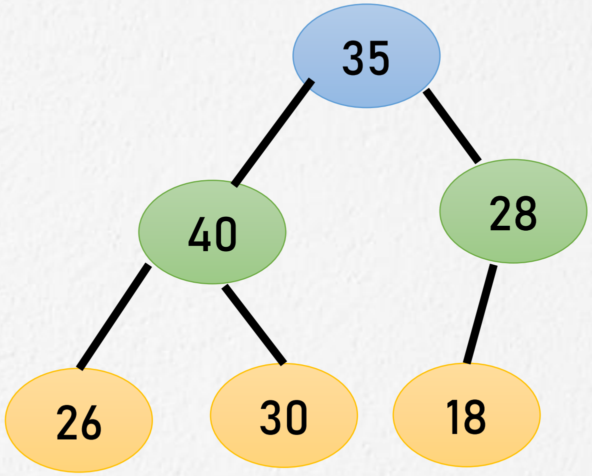


Step 5

Heap Tree construction

Data: 35 30 18 26 40 28 10



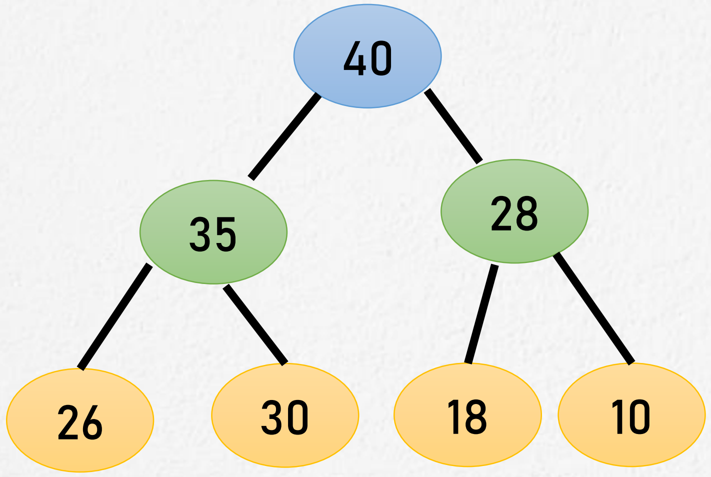


Step 6

Step 7

Heap Tree construction

Data: 35 30 18 26 40 28 10

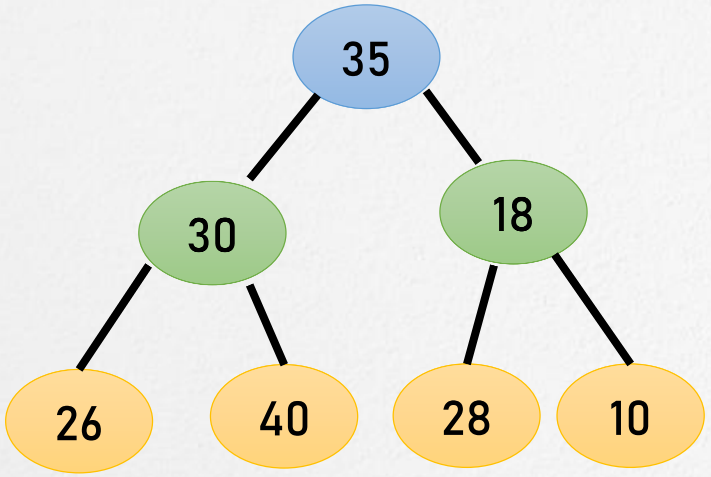


Step 8

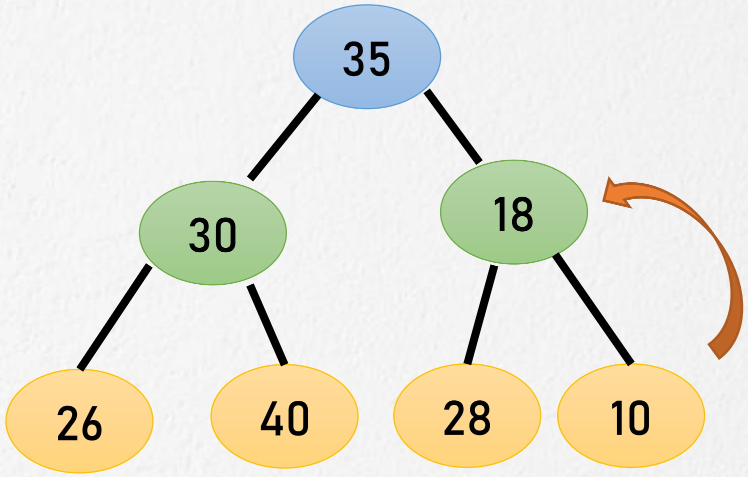
Heapify Method (Min Heap)

Data: 35 30 18 26 40 28 10

Complexity: O(n)

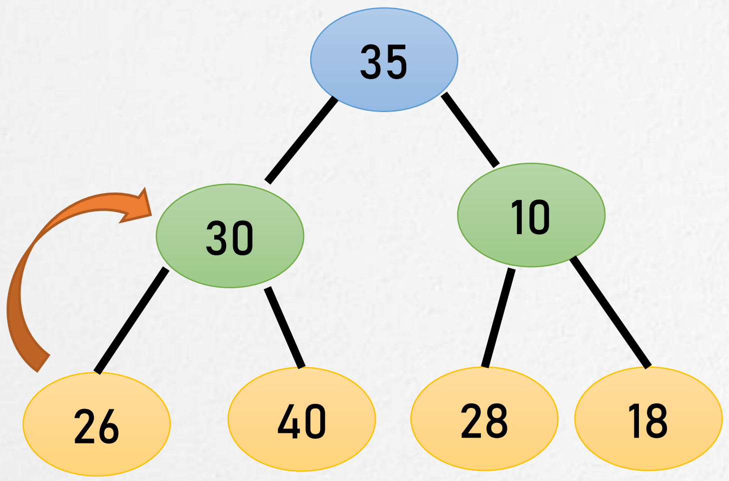


Step 1

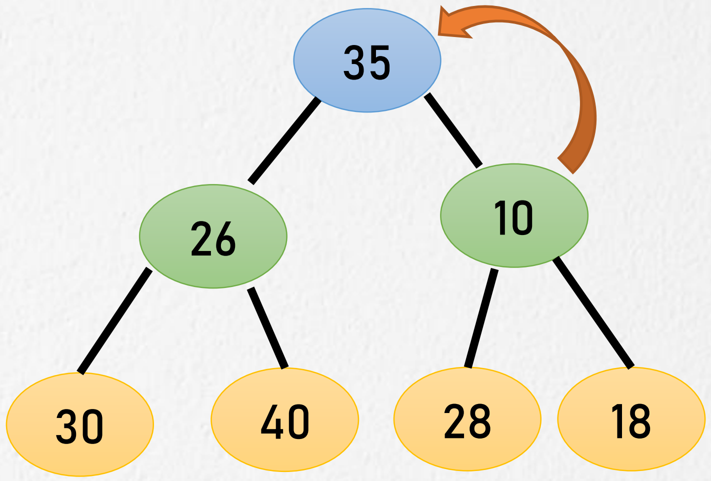


Step 2

Heapify Method (Min Heap)

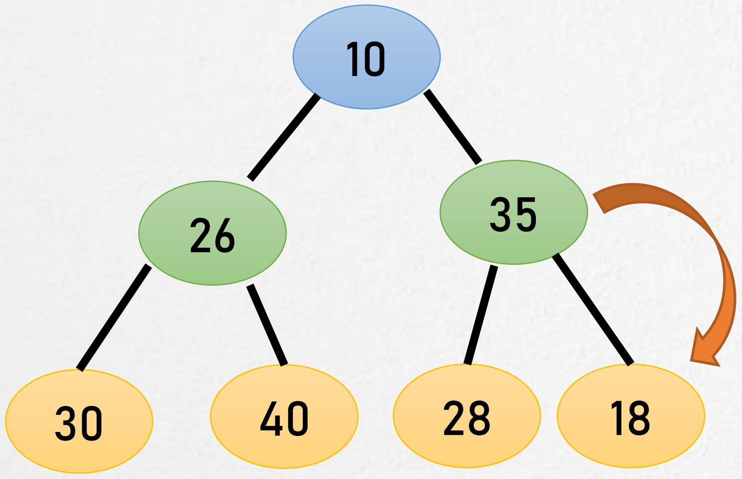


Step 3

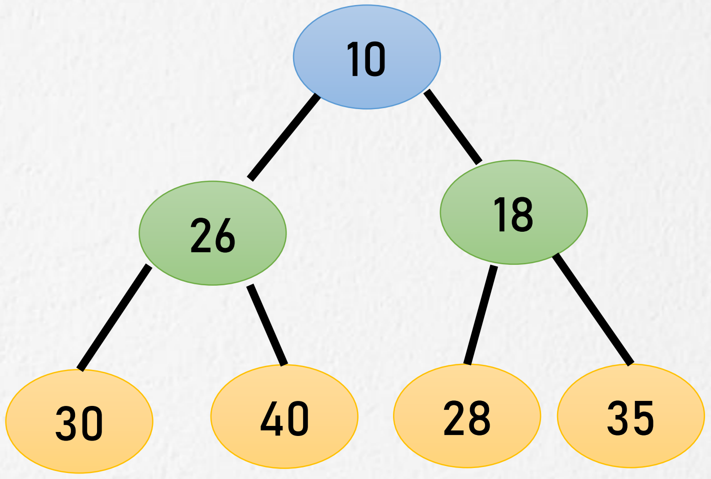


Step 4

Heapify Method (Min Heap)



Step 5



Step 6

Deletion: Heap Tree (01)

1. Remove root node.

2. Move the last element of last level to root.

3. Compare the value of this child node with its parent.

4. If value of parent is less than child, then swap them.

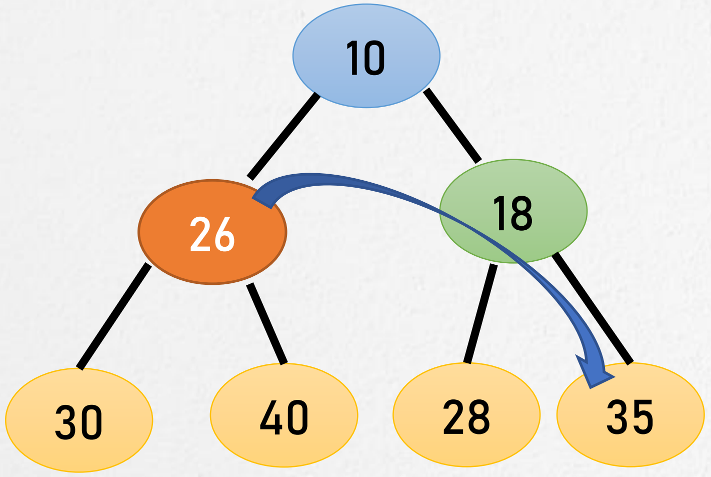
5. Repeat step 3 & 4 until Heap property holds.

Deletion: Heap Tree (02)

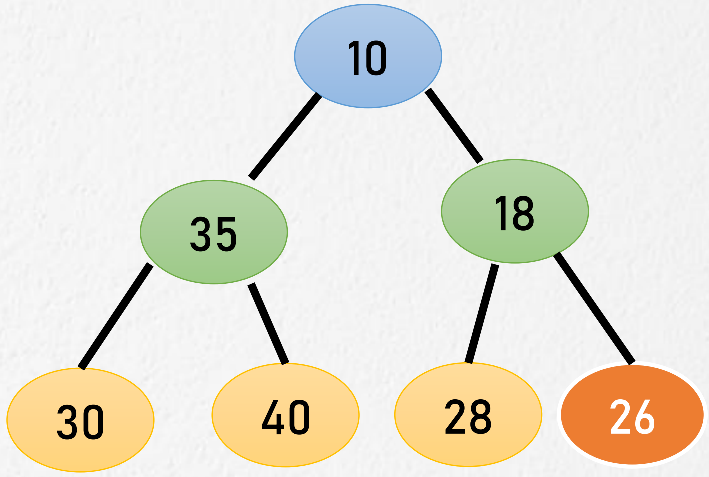
1. Select the element to be deleted.
2. Swap it with the last element.
3. Remove the last element.
4. Heapify the tree.

Deletion: Heap Tree

Element for deletion: 26

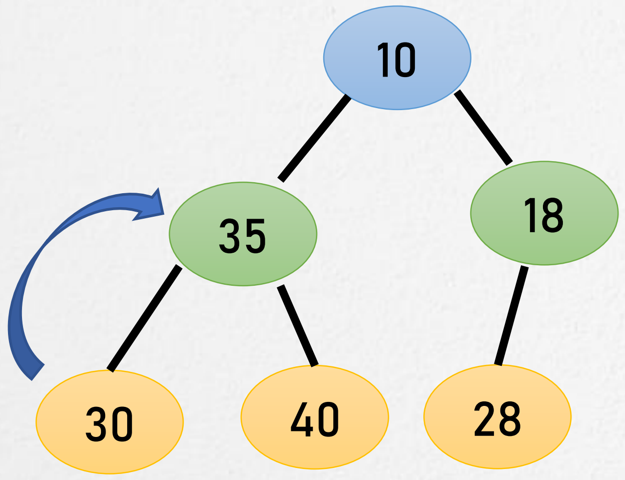


Step 1

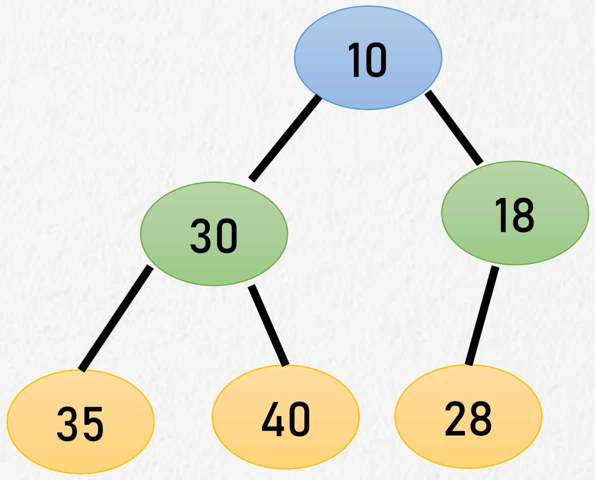


Step 2

Deletion: Heap Tree



Swap the element



Heapify the tree

That's all for now…