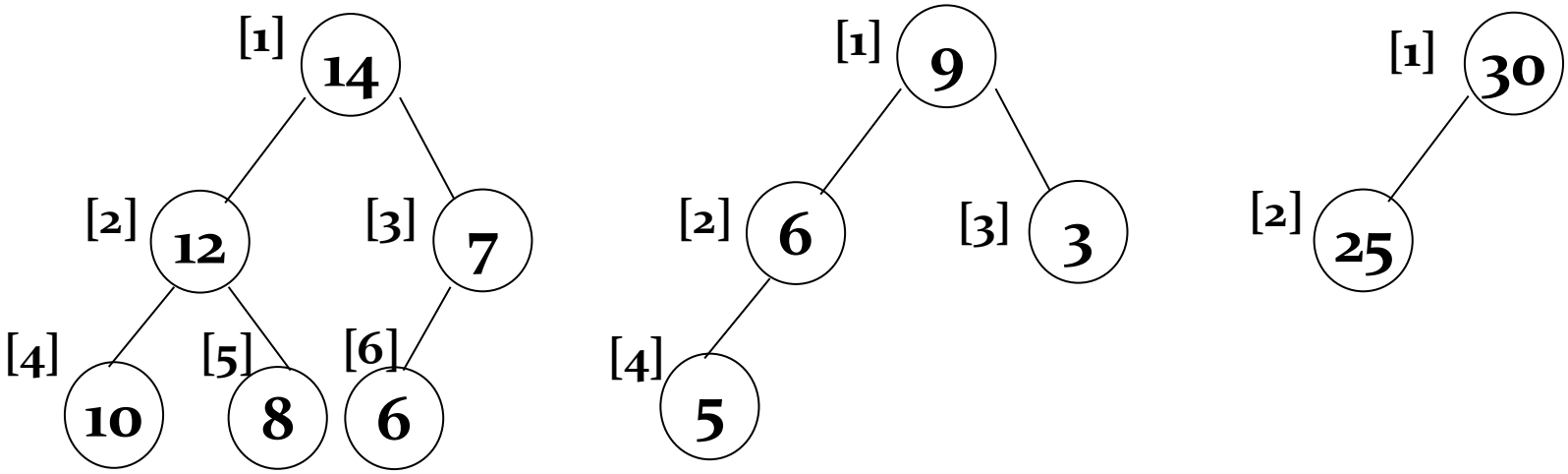
Heaps-Trees

- insertion,
- deletion,
- sorting and
- complexity analysis

Heap

- A *max tree* is a tree in which the key value in each node is no smaller than the key values in its children. A *max heap* is a complete binary tree that is also a max tree.
- A *min tree* is a tree in which the key value in each node is no larger than the key values in its children. A *min heap* is a complete binary tree that is also a min tree.
- Operations on heaps
  - creation of an empty heap
  - insertion of a new element into the heap;
  - deletion of the largest element from the heap

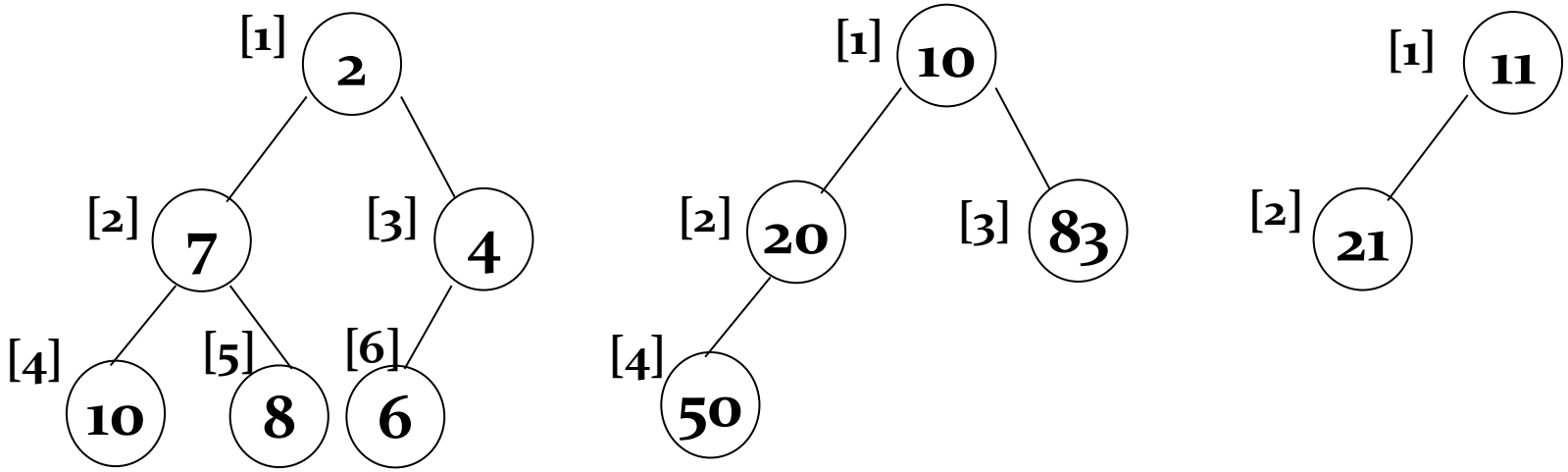
Figure : Sample max heaps



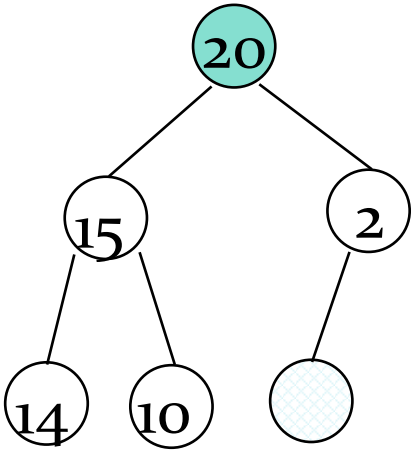
Property:

The root of max heap (min heap) contains the largest (smallest).

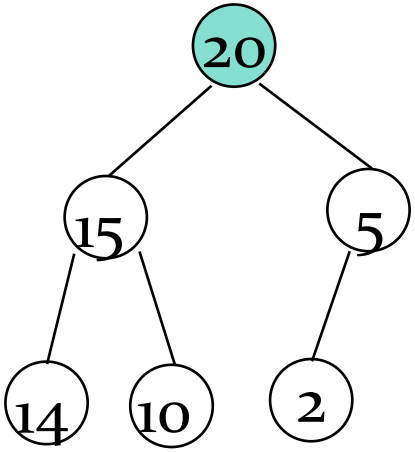
Figure : Sample min heaps



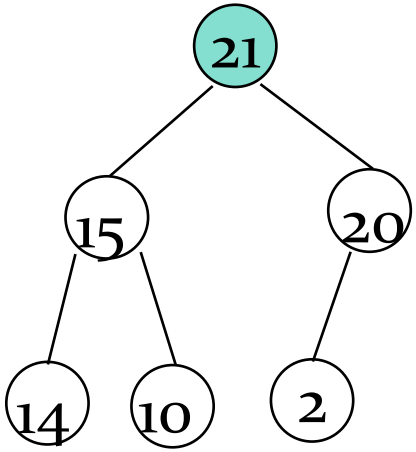
Example of Insertion to Max Heap



initial location of new node



insert 5 into heap



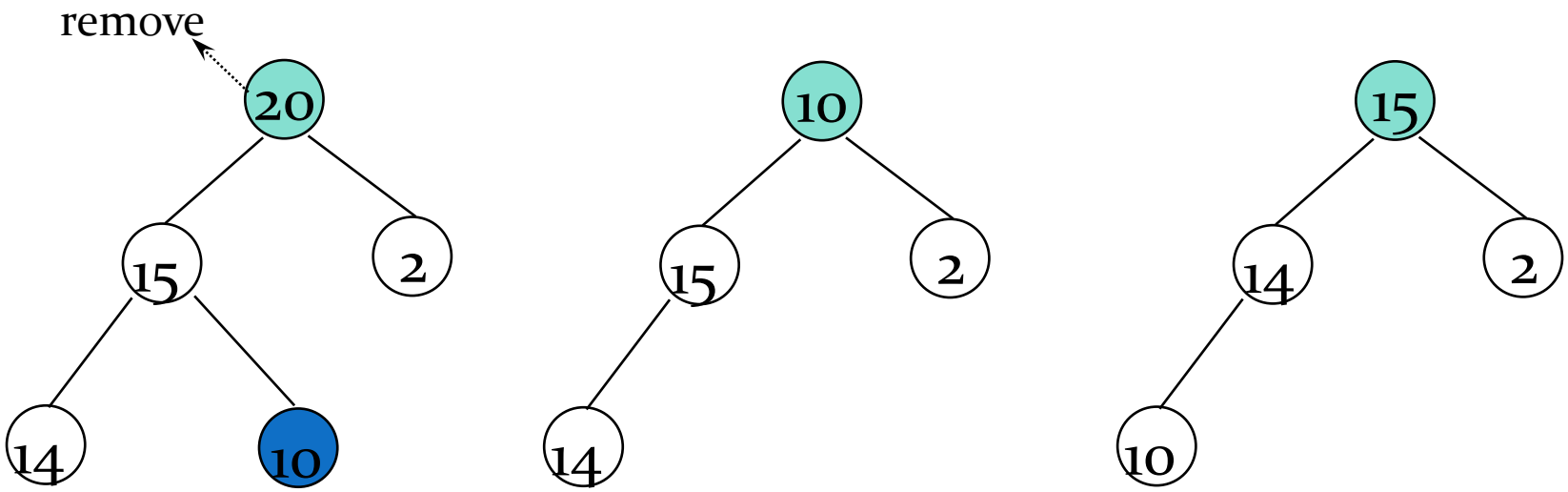
insert 21 into heap

Insertion into a Max Heap

INSHEAP(TREE, N, ITEM)

1. Set N=N+1 and PTR=N.
2. Repeat Steps 3 to 6 while PTR>1.
3. Set PAR= ⌊PTR/2⌋
4. If ITEM<= TREE[PAR], then:
   Set TREE[PTR]=ITEM, and Return.
   [End of If structure]
5. Set TREE[PTR]=TREE[PAR]
6. Set PTR=PAR.
   [End of step 2]
7. Set TREE[1]=ITEM
8. Return.

Example of Deletion from Max Heap



Deletion from a Max Heap

DELHEAP(TREE, N, ITEM)

1. Set ITEM= TREE[1].
2. Set LAST= TREE[N] and N= N-1.
3. Set PTR=1, LEFT=2 and RIGHT=3.
4. Repeat Steps 5 to 7 while RIGHT<=N:
5. If LAST>=TREE[LEFT] and LAST>= TREE[RIGHT], then:

   Set TREE[PTR]=LAST and Return.

   [End of If structure.]
6. If TREE[RIGHT]<= TREE[LEFT], then:

   Set TREE[PTR]= TREE[LEFT] and PTR=LEFT.

   Else:

   Set TREE[PTR]= TREE[RIGHT] and PTR=RIGHT.

   [End of If structure.]
7. Set LEFT=2*PTR and RIGHT=LEFT+1.

   [End of Step 4 loop.]
8. If LEFT=N and if LAST < TREE[LEFT] set TREE[PTR]= TREE[LEFT]

   Set PTR=LEFT.
9. Set TREE[PTR]= LAST.
10. Return.

HeapSort

HEAPSORT(A, N)

1. Repeat for J=1 to N-1:

   Call INSHEAP(A, J, A[J+1]).

   [End of loop.]

2. Repeat while N>1:

   a) Call DELHEAP(A, N, ITEM).

   b) A[N+1]=ITEM.

   [End of loop.]

3. Exit.

Thank You