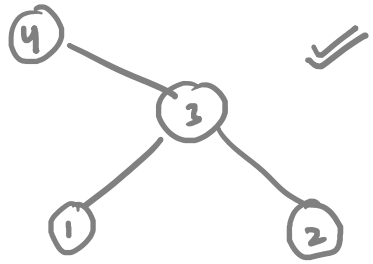
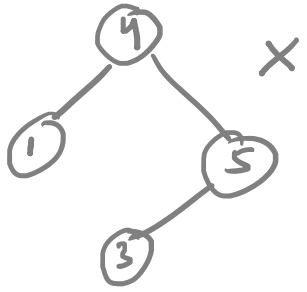
# Heap

Wednesday, October 20, 2021 10:40 AM

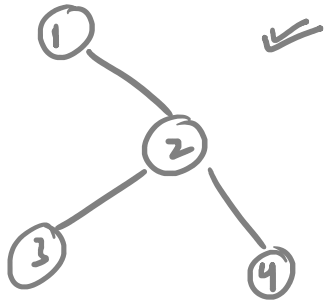
**Heap :-** A Complete binary tree which is either a max tree or a min tree.

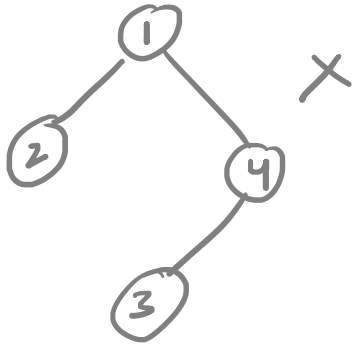
→ Max Tree – Value of every node is greater than all its descendants.



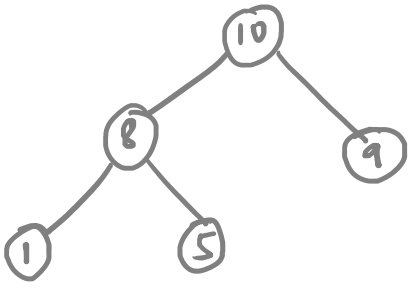




→ Min Tree – Value of every node is smaller than all its descendants.

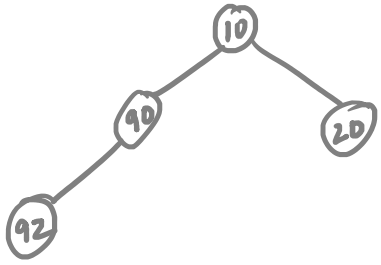


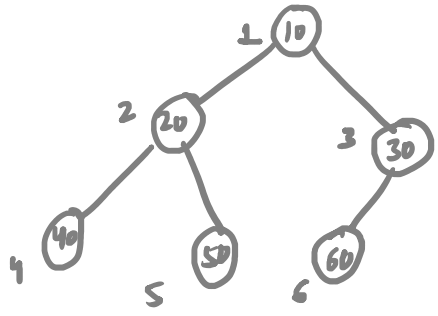

→ Max-Heap – A max tree which is Complete.

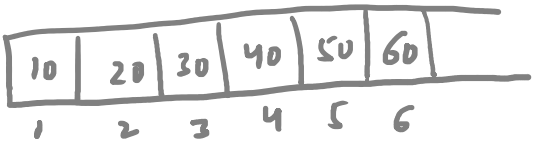



→ Min Heap – A min tree which is Complete.




## Array based representation of a Heap




Array A

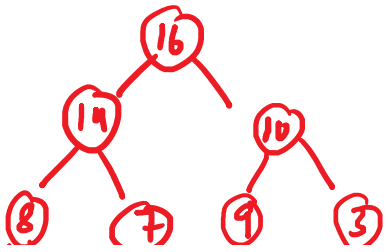
1) Root node ← A[1]
2) Parent of A[i] ← A[⌊i/2⌋]
3) Left child of A[i] ← A[2i]
4) Right child of A[i] ← A[2i+1]

## Applications of Heap:-

1) Priority Queue
2) Sorting (Heapsort)

## Heap as a Priority Queue:-

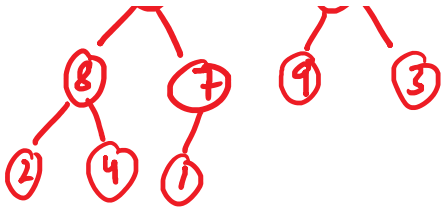
Consider MAX-Heap





Always the root is at the highest priority

Let the root's info is stored to S.

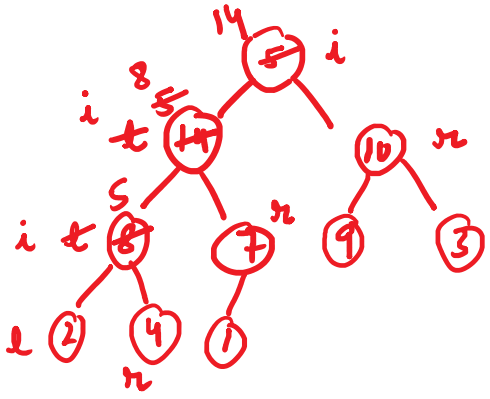


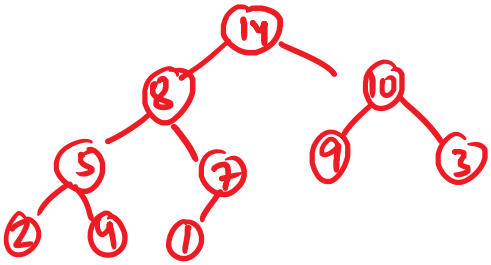
priority

Let say root's info is changed to 5.

MAX-HEAPIFY (A, n, i)

1) $l \leftarrow 2i$
2) $r \leftarrow 2i+1$
3) largest $\leftarrow i$
4) if $l \leq n$ && $A[l] > A[i]$
   largest $\leftarrow l$
5) if $r \leq n$ && $A[r] > A[largest]$
   largest $\leftarrow r$
6) if largest $\neq i$
   swap $A[i] \leftrightarrow A[largest]$
   MAX-HEAPIFY (A, n, largest)





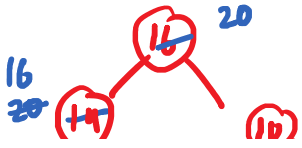
$\theta(\log_2 n)$

$O(\log_2 n)$

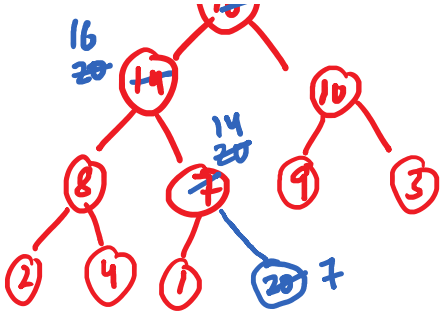
## Priority Queue

1) Insertion :- INSERTION (A, n, x)

1) $n \leftarrow n+1$
2) $A[n] \leftarrow x$
3) $i \leftarrow n$
4) while ( $i > 1$ && $A[i] > A[\lfloor i/2 \rfloor]$
   swap $A[i] \leftrightarrow A[\lfloor i/2 \rfloor]$
   $i \leftarrow \lfloor i/2 \rfloor$



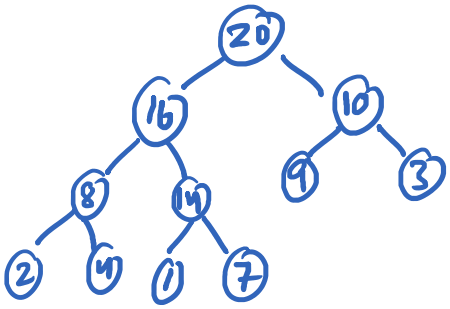
Insert (A, 10, 20)



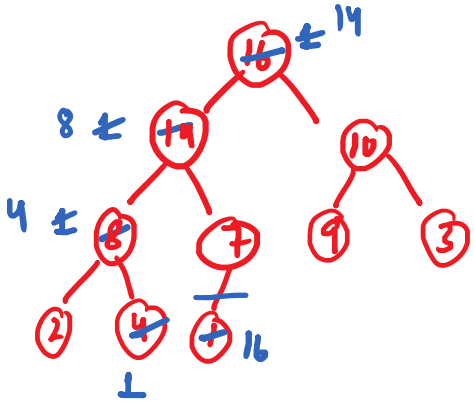
$\theta(\log_2 n)$

$O(\log_2 n)$

Insert (A, 10, 20)

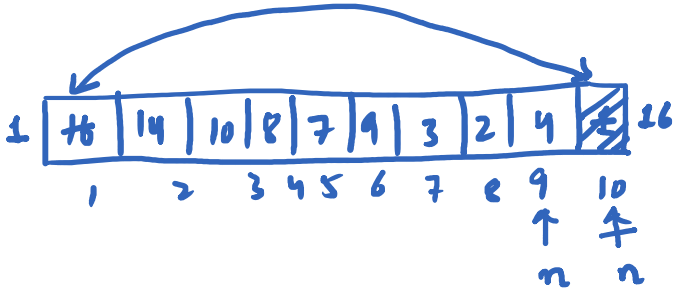


2) Deletion :-



$\theta(\log_2 n)$

$O(\log_2 n)$



DELETION (A, n)

1) Swap $A[1] \leftrightarrow A[n]$

2) $n \leftarrow n-1$

3) MAX-HEAPIFY (A, n, 1)

Heapsort :-

Selection of Heap → Ascending order — MAX Heap

Selection of Heap → Descending order — MIN Heap

HEAPSORT (A, n)

1) for $i \leftarrow \lfloor n/2 \rfloor$ down to 1

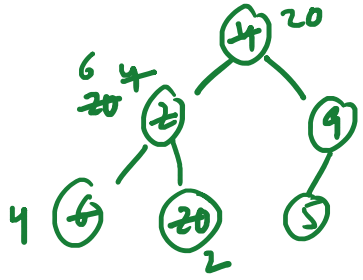
MAX-HEAPIFY (A, n, i)

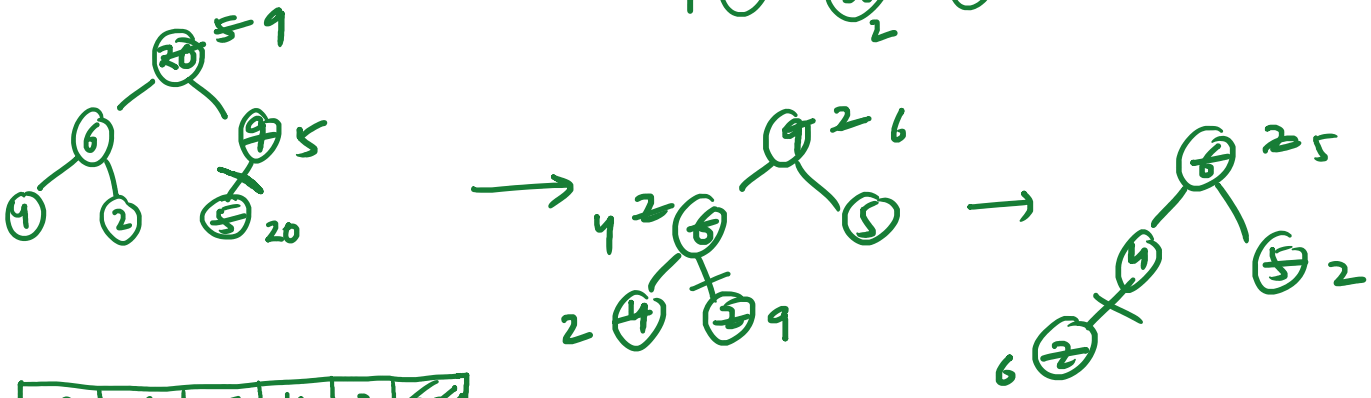
$\theta(n \log_2 n)$

$O(n \log_2 n)$

2) for i ← n downto 2
   swap A[1] ↔ A[i]
   MAX-HEAPIFY (A, i, 1)

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| 4 | 2 | 9 | 6 | 20 | 5 |





| | | | | | |
|---|---|---|---|---|---|
| 9 | 6 | 5 | 4 | 2 | 20 |
| 6 | 4 | 5 | 2 | 9 | 20 |
| 5 | 4 | 2 | 6 | 9 | 20 |
| 2 | 4 | 5 | 6 | 9 | 20 |
| 2 | 4 | 5 | 6 | 9 | 20 |