mf → {}

linked list
arrays

Priority Queue

BST

fcfs

fifo

Normal → el1 el2 → the element that came first could be picked

Priority → max
→ min
→ price * 0.7 + rating * 0.3

LIFO

insertion

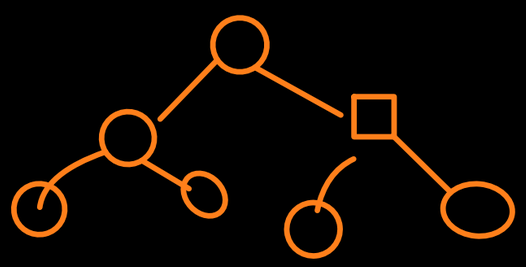
O(n)

O(1)

O(h)

| 3 | 8 | 1 | 5 | 4 |

→ sorted → any



| 1 | 3 | 4 | 5 | 8 | X |

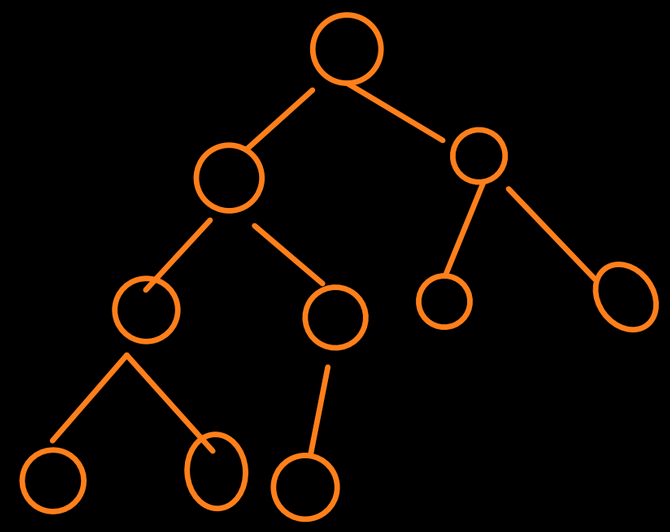
# Heaps → using heaps we can implement PQ

## Facts

↳ Heap is a binary tree

↳ Heap is a complete binary tree

↳ Heap will maintain a mental model of Complete B.T but will be very easy implemented using arrays.

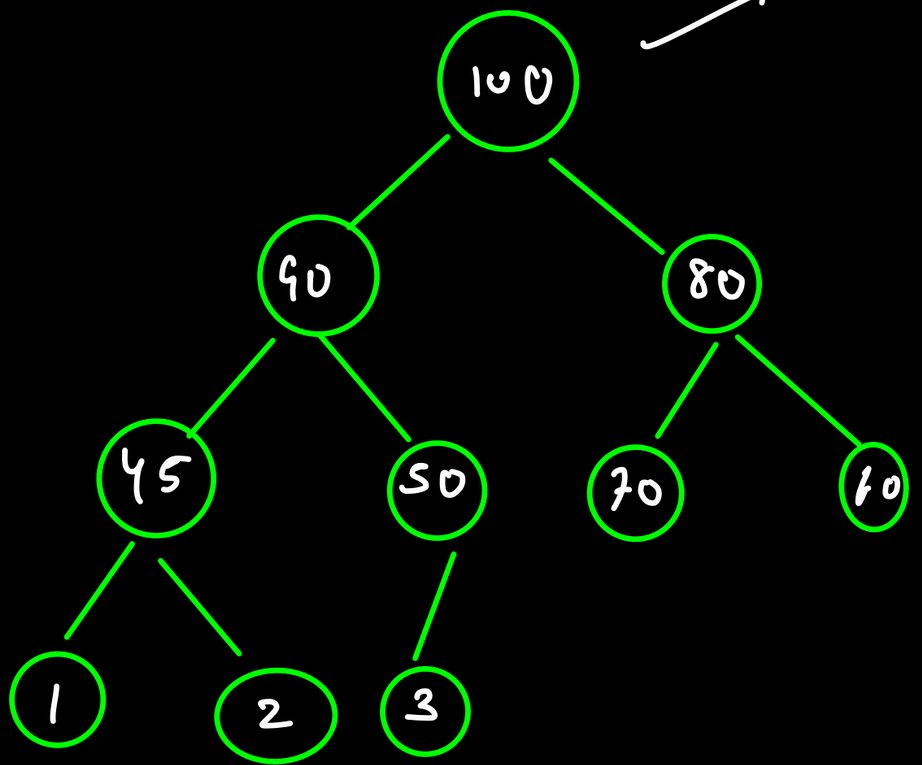


↳ as this is a tree, parent has Higher Priority than the children.

Priority → max (Max heap)

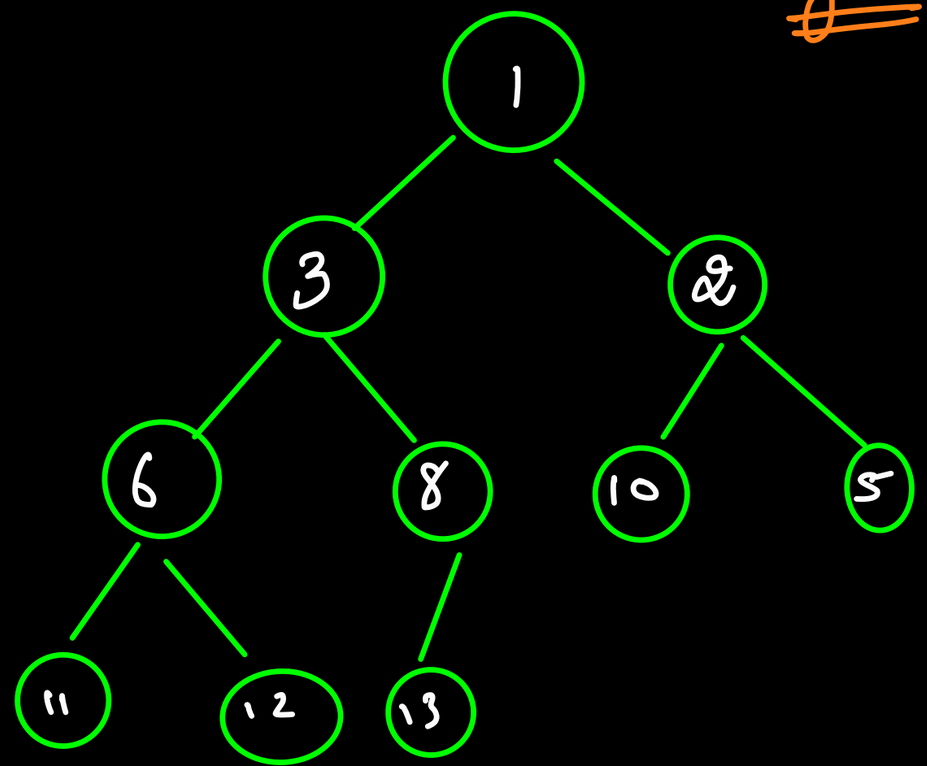
Priority → min (Min heap)

Max heap → root → is the highest priority element




Min heap

get → O(1)



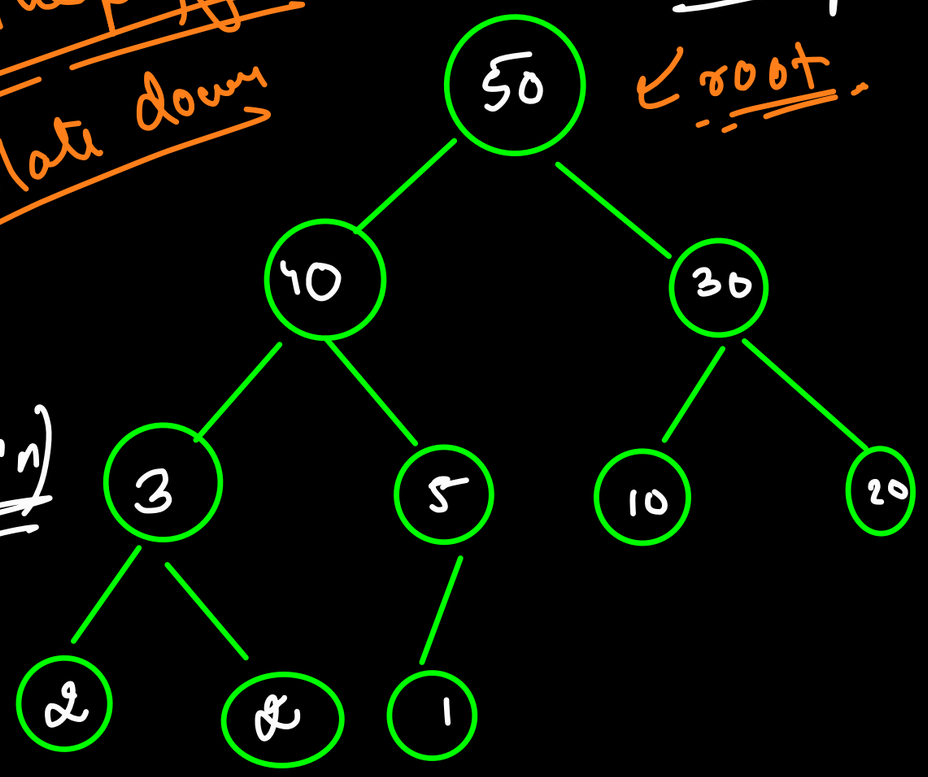
downheapify → O(h)

percolate down

O(log n)

Max heap

root

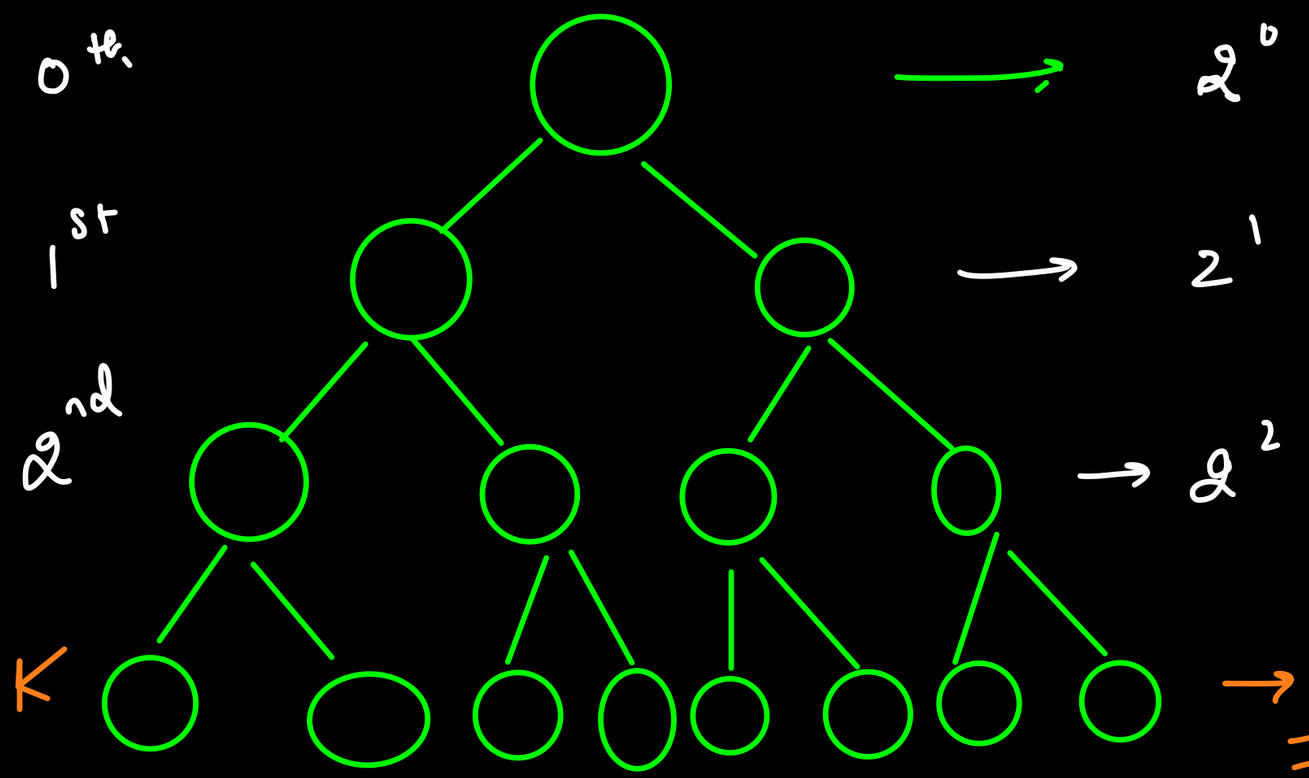



for a heap to be valid, every subtree should also be a heap.

$$Priority_{parent} > Priority_{child}$$

is this a max heap

Q- Suppose your lst & rst are max heap but not the whole tree, how to resolve it?




worst case CBT can be perfect.

So the last level has $2^k$ nodes

Height → (K+1)

Total n node → $\underbrace{2^0 + 2^1 + 2^2 + 2^3 \cdots\cdots 2^{K-1}}_{\text{K}} + \underbrace{2^K}_{\text{no. of node on last level}} = n$

$\frac{a(r^n - 1)}{r-1}$ → $\frac{2^0 \times (2^K - 1)}{2-1}$ ⇒ $(2^K - 1)$

$2^K - 1 + 2^K = n$ → $2 \times 2^K = n+1$ ⇒ $2^K = \frac{n+1}{2}$

$$2^k = \frac{n+1}{2}$$ Taking $\log_2$ both sides

$$\log_2 2^k = \log_2\left(\frac{n+1}{2}\right)$$

$$\log_2 2^k = \log_2(n+1) - \log_2 2$$

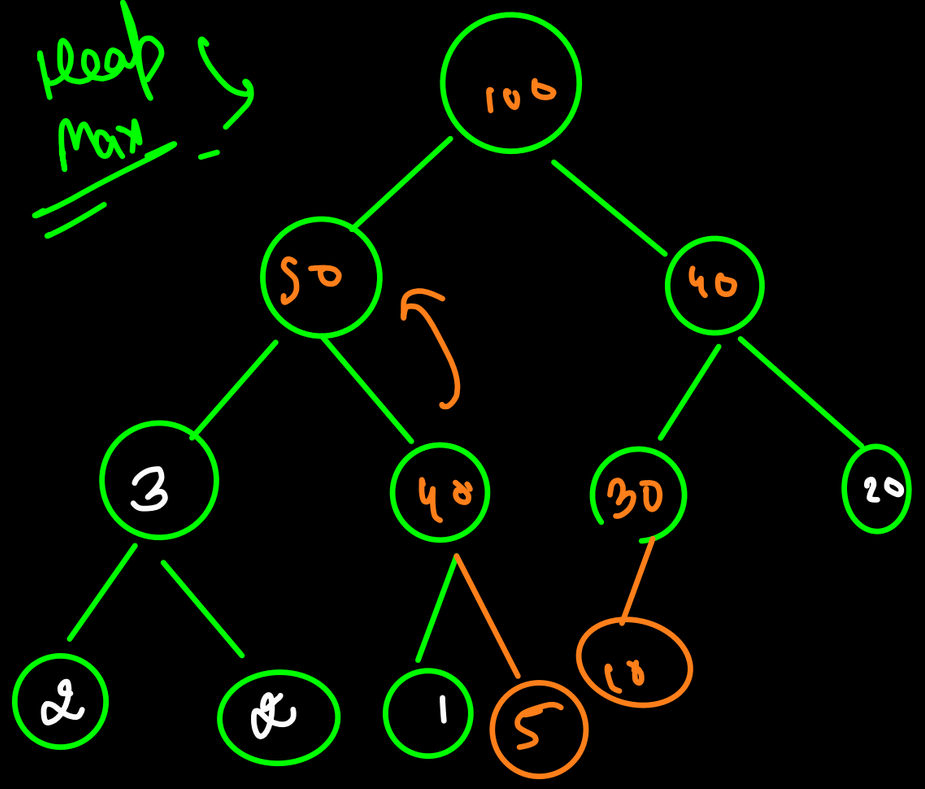
$$k \log_2 2 = \log_2(n+1) - 1$$

$$k = \log_2(n+1) - 1$$

$$k+1 \Rightarrow \log_2(n+1)$$

$$k \rightarrow O(\log n)$$

Heap
Max




insertion → O(log n)

100

40

upheapify

percolate up

↓

O(h)

↓

O(log n)

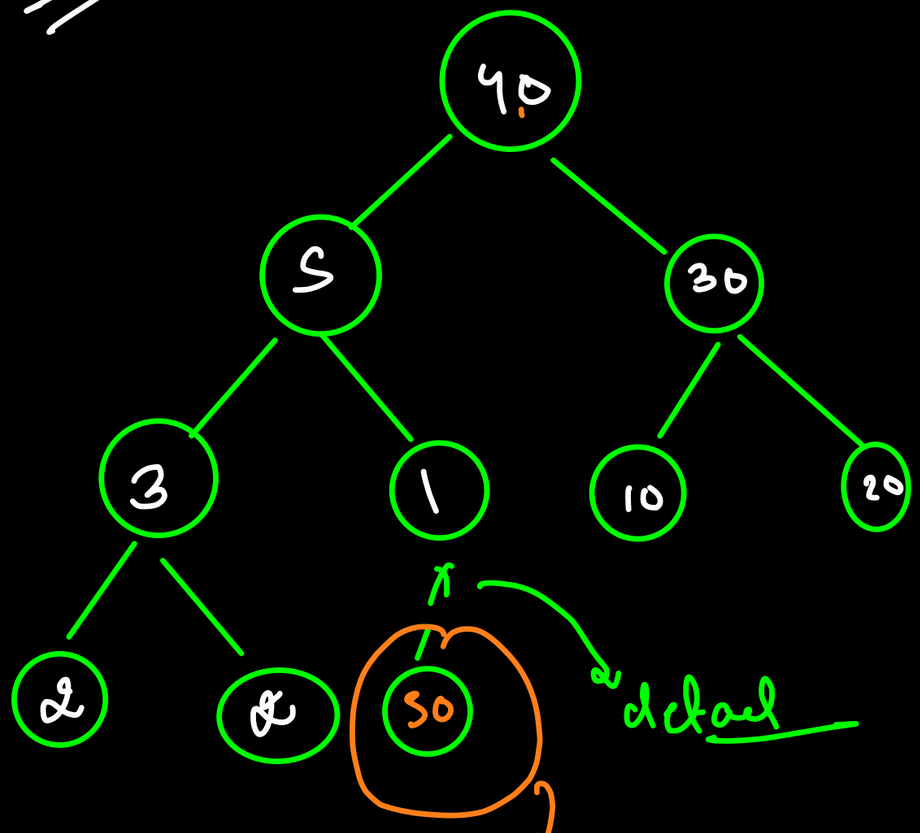
$10^9$

$\log 10^9 \approx 30$

downheapify

How can we remove the highest priority element ??

→ O(log n)

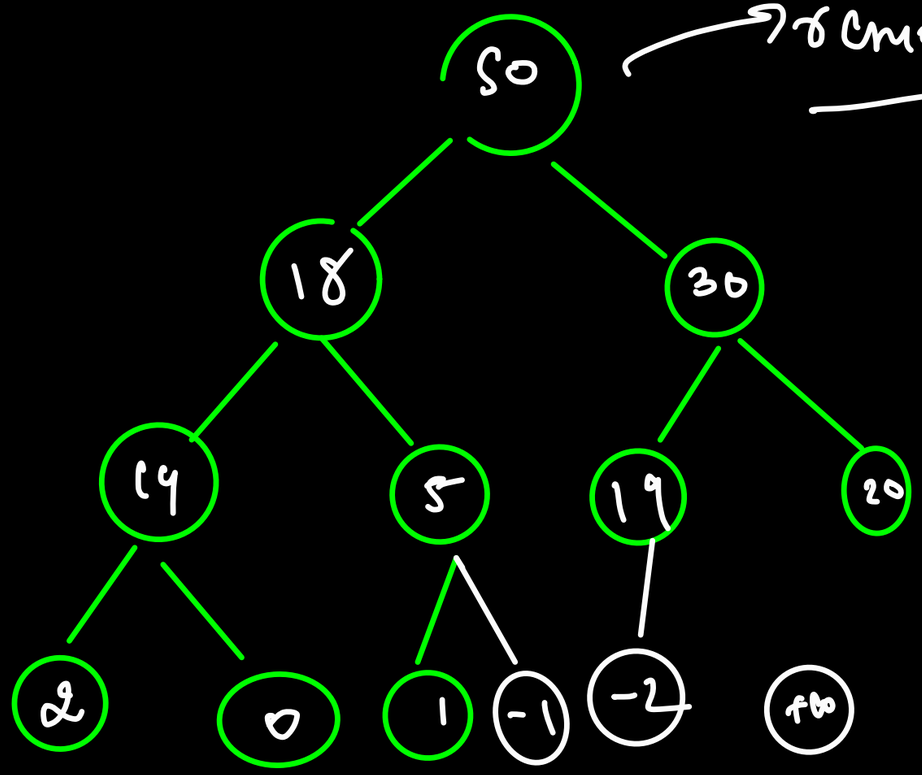



Swap the root with the last element

safest element to remove, as this doesn't hamper complete Binary tree property

How can we remove any element of the heap (not just the root??)

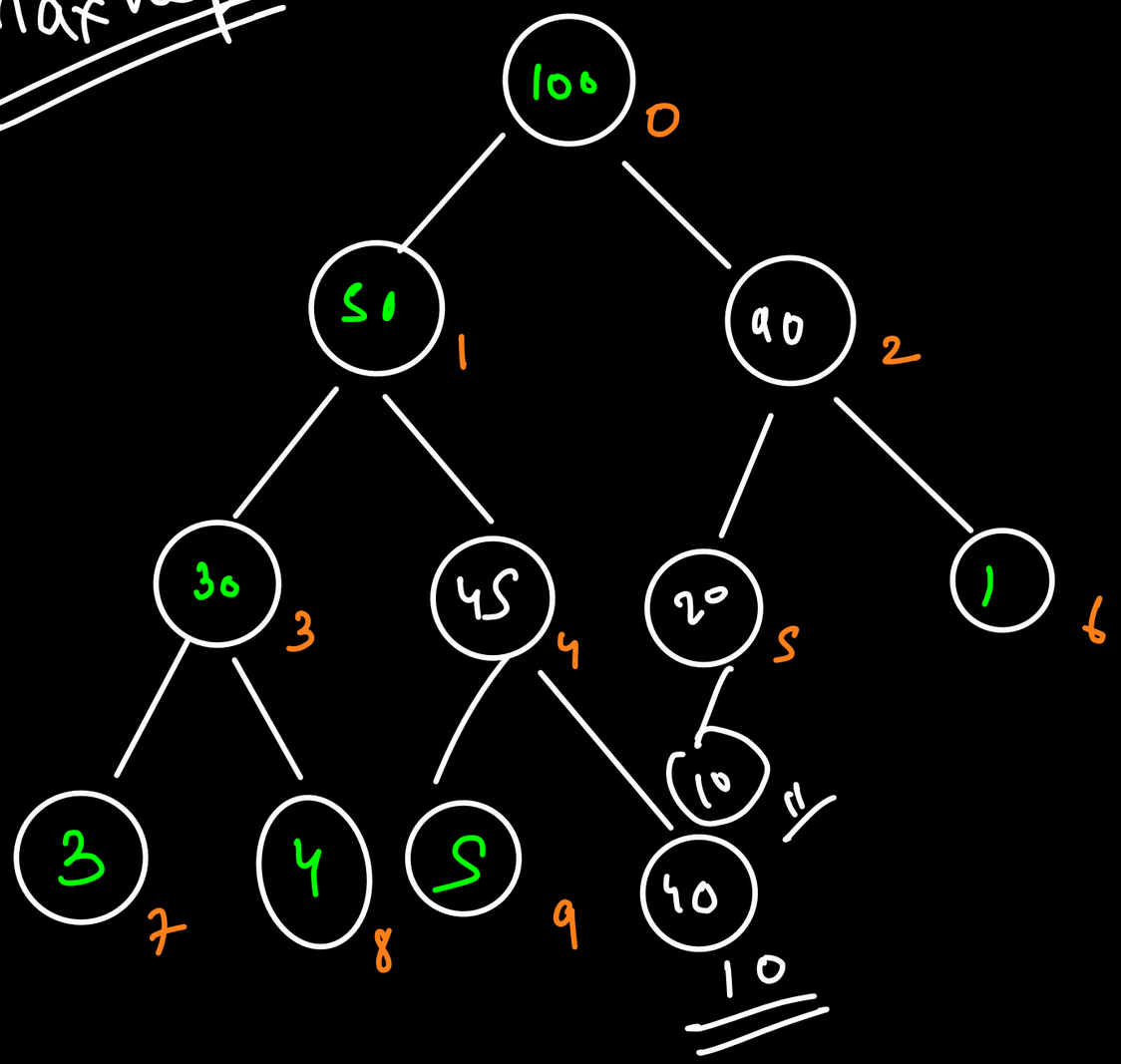
Naive



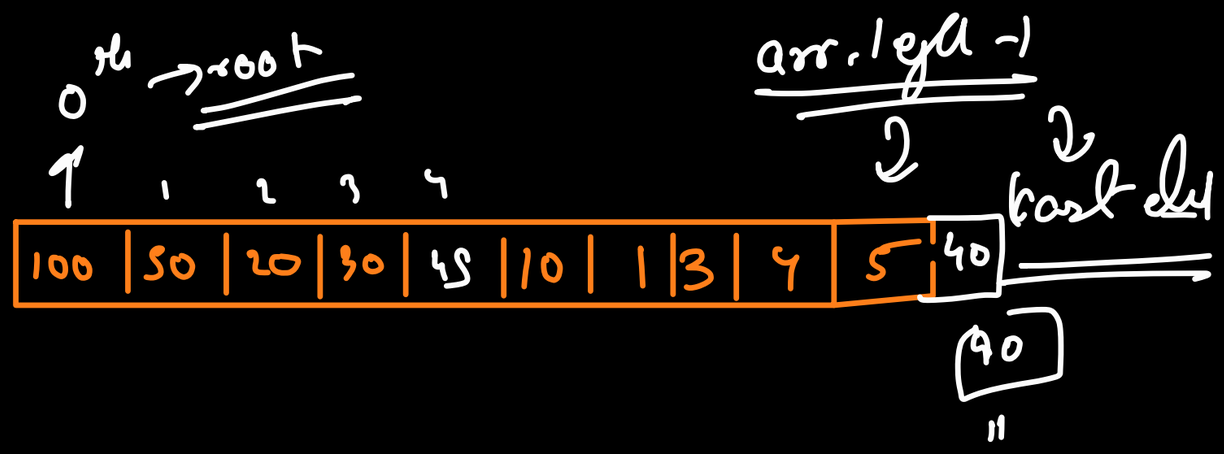

$O(\log n + \log n)$

$O(\log n)$

# Max heap




idx = 2
pi = 0




left child $\Rightarrow 2 \times pi + 1$

right child $\Rightarrow 2 \times pi + 2$

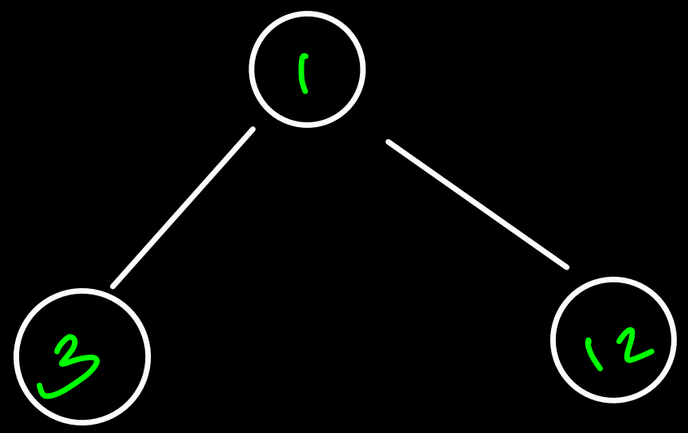
$$pi \Rightarrow floor\left(\frac{(child - 1)}{2}\right)$$

```
upheapify (heap, i) {
    pi = Math.floor( (i-1)/2 )
    while (pi >= 0) {
        if (heap[i] > heap[pi])
            swap [heap[i], heap[pi])
        pi = Math.floor( (pi-1)/2 );
    }
}
```

i = 10

pi = 4 ~~1~~ 0

4-1/2 1-1/2 . 0




result = 1 3 12