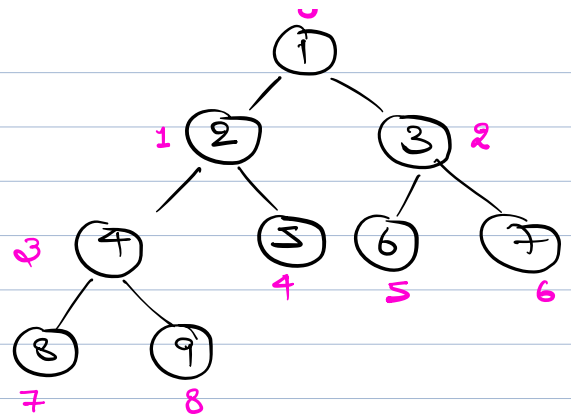


1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8

Parent	LC	RC
0	1	2
1	3	4
2	5	6
3	7	8

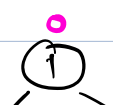
Index of parent (i)

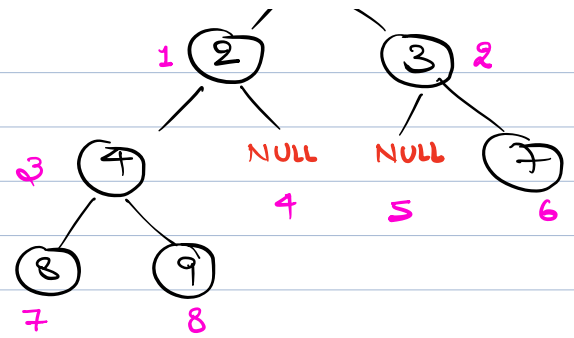
↳ Left Child = $(2 \times i) + 1$

Right Child = $(2 \times i) + 2$

Index of the child = i

↳ Parent = $(i-1)/2$





1	2	3	4	NULL	NULL	7	8	9
0	1	2	3	4	5	6	7	8

Q N ropes with their length.

Cost of connecting 2 ropes

= sum of length of 2 ropes.

find the minimum total cost to connect all the ropes.

Eg A: [2, 5, 2, 6, 3]

1) $2 + 5 = 7 + 2 = 9 + 6 = 15 + 3 = 18$

$7 + 9 + 15 + 18 = 49$

2) $2 + 2 = 4$ $\quad$ $5 + 6 = 11$

$4 + 3 = 7$ $\quad$ $7 + 11 = 18$

$$4 + 11 + 7 + 18 = \underline{\underline{40}}$$

A: [~~5~~, ~~6~~, 7, 8, 9, 11

11

$O(N)$

$\downarrow$

$O(N^2)$

Intuition

$x, \quad y, \quad z \qquad x < y < z$

1) $x + y$

2) $(x+y) + z$

$\underline{\qquad\qquad}$

1 + 2

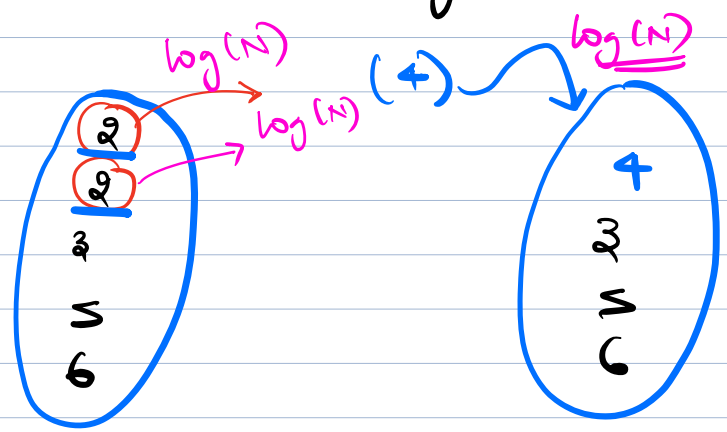
$>$ ($x + z$, $(x+z) + y$; $y + z$, $(y+z) + x$)

2 is same for all 3 cases.

$\Rightarrow$ At any point of time, out of all the available choices, select 2 ropes with min possible length

$$O(N \log N + N^2) = O(N^2)$$

⇒ Let's suppose we have a DS. where we can insert an element & remove the min of all the elements in log(N) time.



N log N

1 → log(N)

N → N log(N)

Magical DS ⇒ Heap

Binary Heap

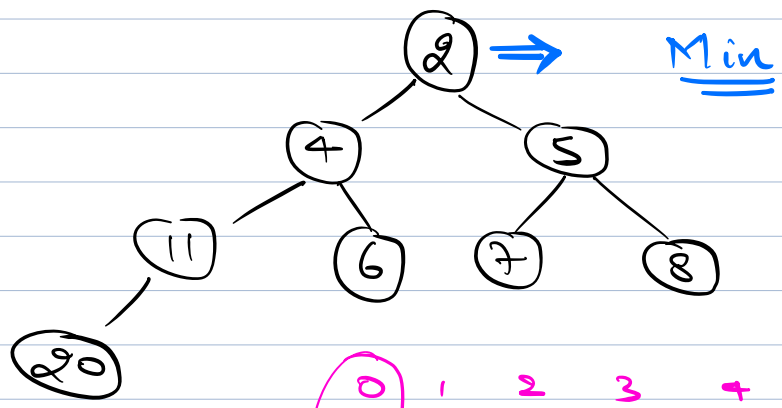
Properties : 1) Structure is a Complete Binary Tree.

2) Types → Max Heap
→ Min Heap.

3) Min Heap

$\forall$ nodes ⇒ node.data ≤ node.left.data
& node.data ≤ node.right.data

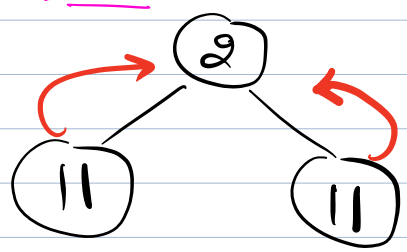
4) There is no relation b/w the left subtree & right subtree of any node.



⇒ Min

0	1	2	3	4	5	6	7
2,	4,	5,	11,	6,	7,	8,	20

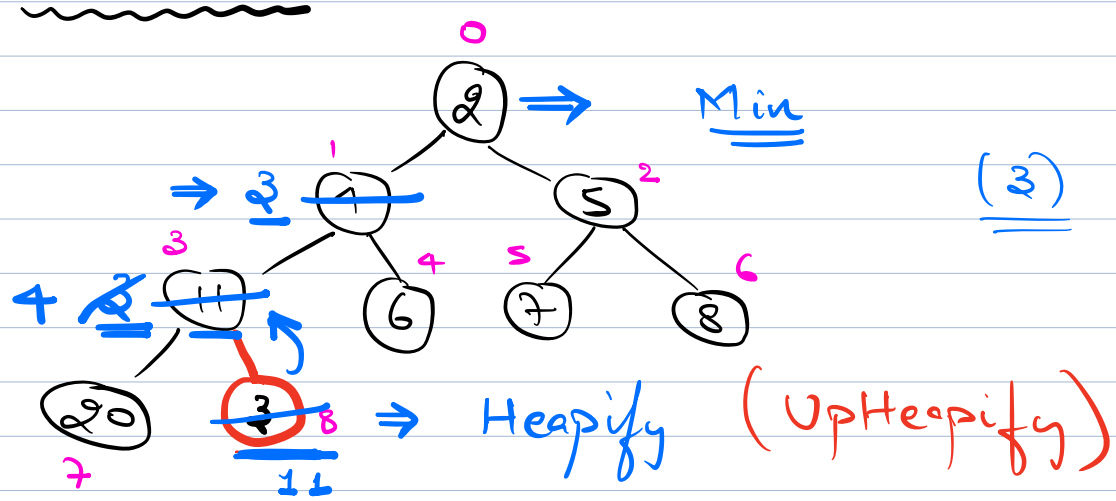
O(1)



Heapify

⇒ Maintain the property of heap. after inserting, or removing.

1) Insertion



2	~~4~~	5	~~11~~	6	7	8	20	~~3~~
0	1	2	3	4	5	6	7	8

Node		Parent		
8	<	(8-1)/2	= 3	swap
3	<	(3-1)/2	= 1	swap
1	>	(1-1)/2	= 0	✓ stop

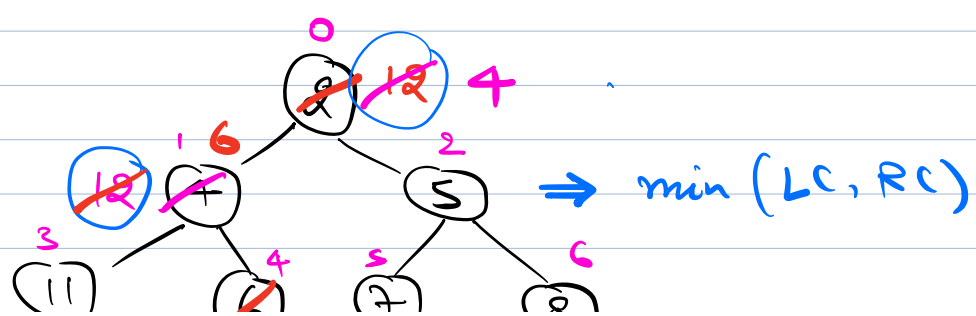
Code

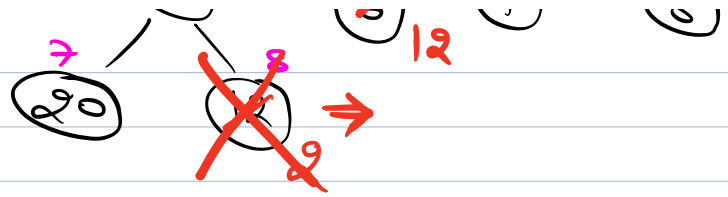
```
void upHeapify ( arr, idx) {
    parent = (idx - 1) / 2
    while ((idx != 0) && ( arr[idx] < arr[parent])){
        swap ( idx, parent);
        idx = parent;
        parent = (idx - 1) / 2
    }
}
```

T.C. = $O(\log N)$

T.C. of insertion = $O(\log N)$

2) Delete Min





Down Heapify

T.C. = $O(\log N)$

Delete Min = $O(\log(N))$

H.W. Code