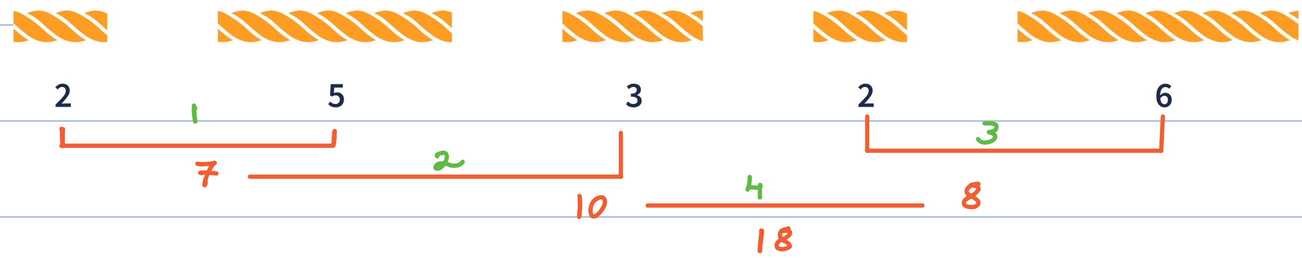
Heaps - 1

TABLE OF CONTENTS

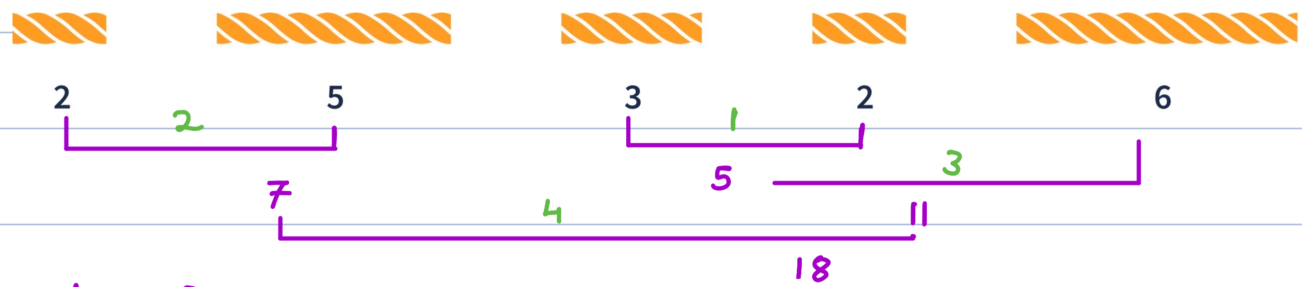
Connecting the Ropes

Given an array that represents the size of different ropes. In a single operation, you can connect two ropes. Cost of connecting 2 ropes is sum of the ropes of length of the ropes you are connecting. Find the minimum cost of connecting all the ropes.

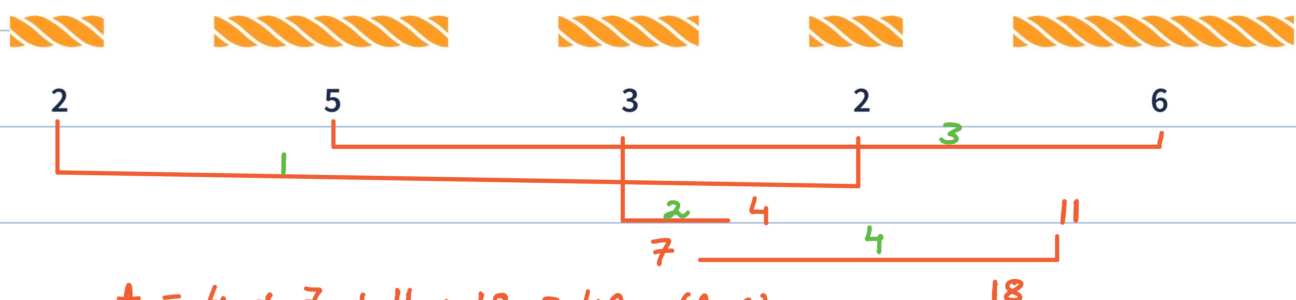
arr[ ] → [2 , 5 , 3 , 2 , 6]



cost = 7 + 10 + 8 + 18 = 43



cost = 5 + 7 + 11 + 18 = 41



cost = 4 + 7 + 11 + 18 = 40 (Ans)

Idea → connect smaller length ropes first.

Let say $x < y < z$.

$x + y$ < $x + z$ < $y + z$

$(x+y)+z$ $\quad$ $(x+z)+y$ $\quad$ $(y+z)+x$

min cost

Sol → Sort in ascending order & travel L → R.

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ {[}5 & 5 & 8 & 11 & 15{]} \end{array}$$

10, 18, ~~X~~

cost = 10 + 18 + . . .

Insertion Sort

TC = $\underline{O(N^2)}$ $\quad$ SC = $\underline{O(1)}$

$$A = \begin{array}{cccc} 0 & 1 & 2 & 3 \\ {[}1 & 2 & 3 & 4{]} \end{array}$$

3, 6, 10

cost = 3 + 6 + 10 = $\underline{19}$

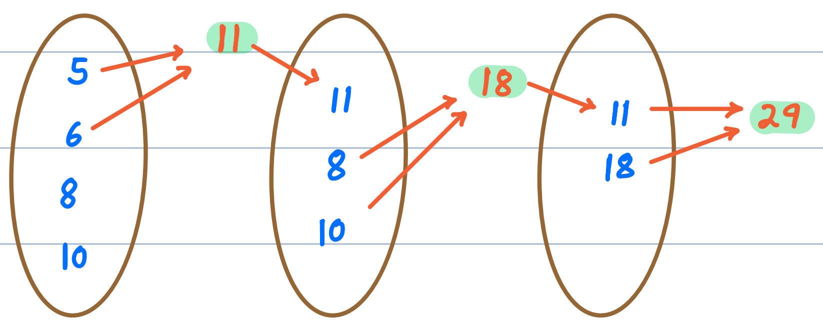
$$A = \begin{array}{cccc} 0 & 1 & 2 & 3 \\ {[}5 & 6 & 8 & 10{]} \end{array}$$

11, 18, 29

cost = 11 + 18 + 29 = $\underline{58}$ (Ans)

Let say, DS → insert element ( ) → get min ( ) } TC = $\underline{O(\log(N))}$

$$A = \begin{array}{cccc} 0 & 1 & 2 & 3 \\ {[}5 & 6 & 8 & 10{]} \end{array}$$



cost = 11 + 18 + 29 = $\underline{58}$

TC = $O((N-1) * 3 \log(N))$

$= \underline{O(N \log(N))}$

SC = $\underline{O(N)}$

Heap Data Structure (a.k.a Priority Queues)

Binary Tree

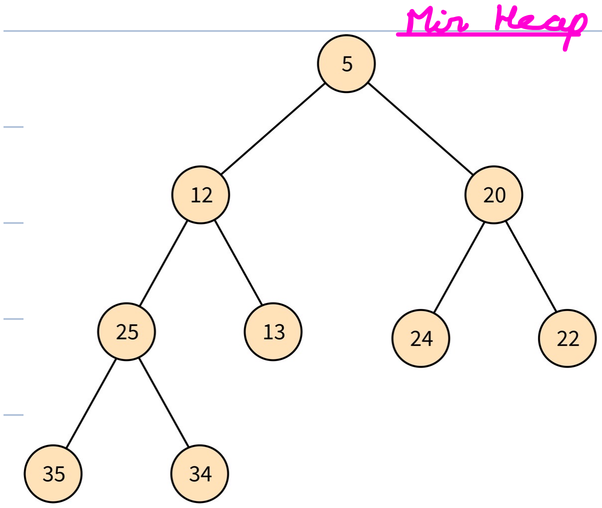
Complete Binary Time (C.B.T) + Heap Order Property (H.O.P)

All levels must be completely filled except possibly the last level and the last level must be filled from left to right.

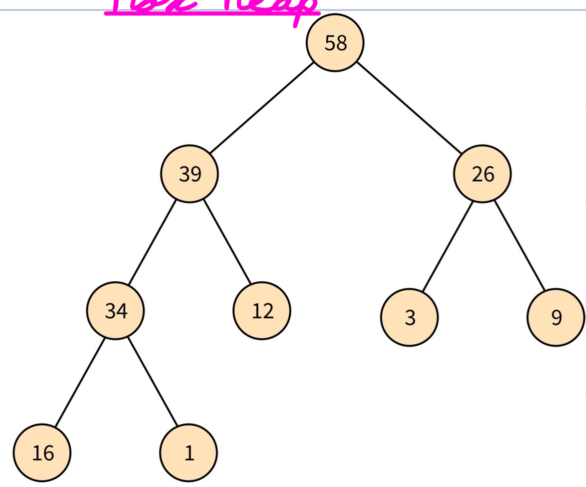
Types → min Heap → ∀ nodes, data <= children data

max Heap → ∀ nodes, data >= children data

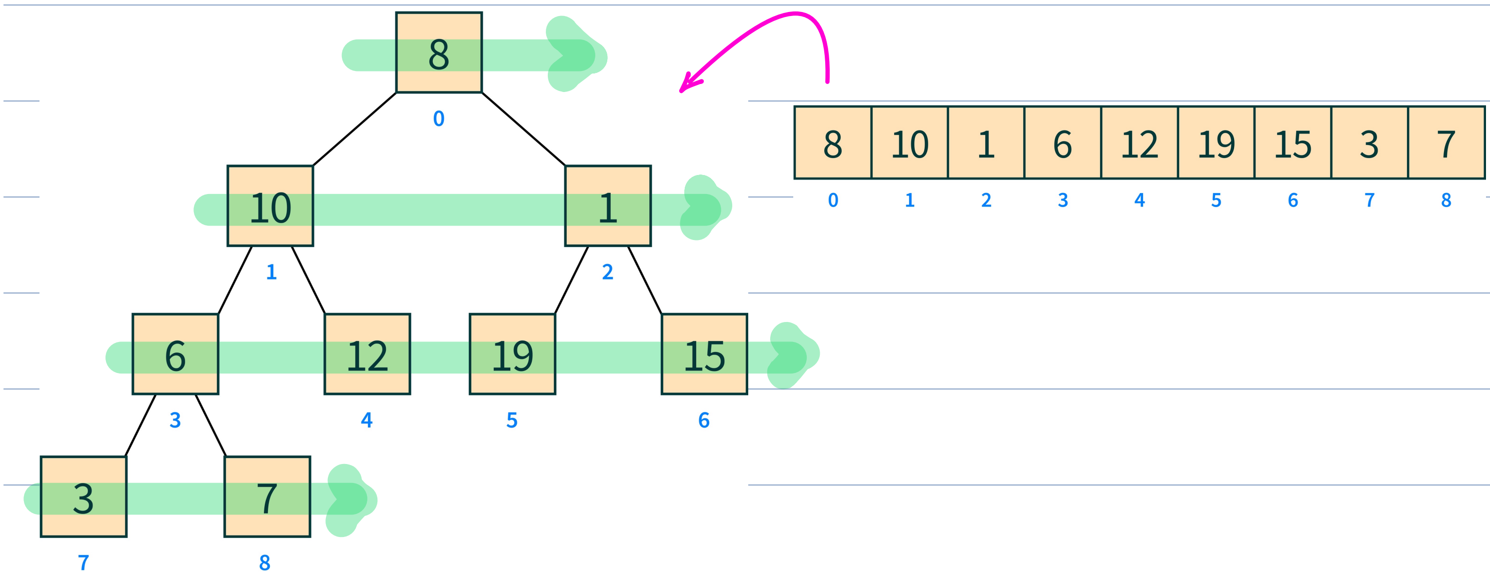
Min Heap



Max Heap



Visualise Arrays as Binary Tree



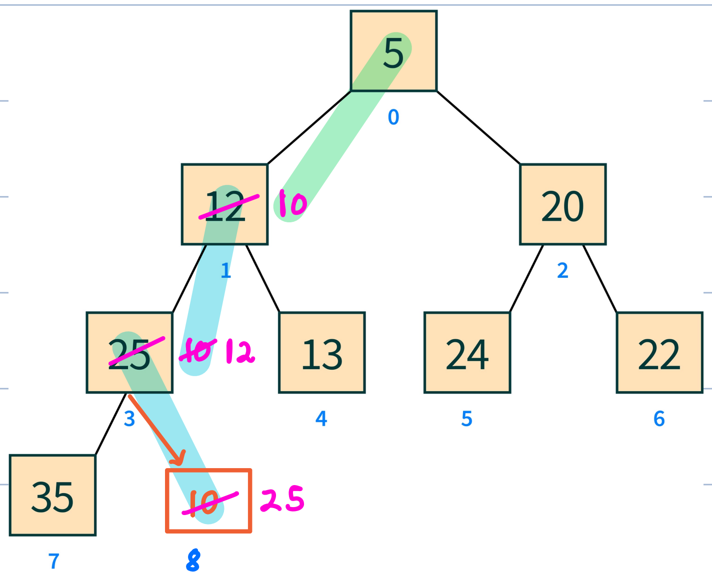
$\forall$ nodes i,

left child $= i * 2 + 1$

right child $= i * 2 + 2$

parent $= (i - 1) / 2$

Min Heap



	10	~~10~~ 12						25
5	~~12~~	20	~~25~~	13	24	22	35	~~10~~
0	1	2	3	4	5	6	7	8

$p = (3-1)/2 = \underline{1}$

$p = (8-1)/2 = \underline{3}$

→ Insert(10) ✓

→ Insert(3) → H.W

∀ nodes, data <= childern data

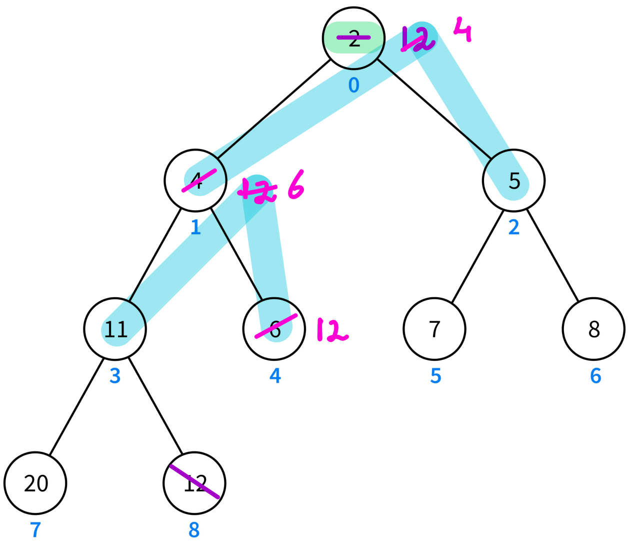
Height of complete binary tree = $\underline{O(\log(N))}$

new insertion

```
A[n] = x                 n++
                         size of heap
i = n
while (i > 0) {
    p = (i - 1) / 2
    if ( A[i] < A[p] ) {
        swap (A, i, p)
        i = p
    } else    break
}
```

TC = $\underline{O(\log(N))}$ SC = $\underline{O(1)}$

Extract-Min / Remove - Min

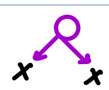


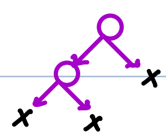
0	1	2	3	4	5	6	7	8
4 ~~12~~ ~~2~~	6 ~~12~~ ~~4~~	5	11	12 ~~6~~	7	8	20	~~12~~

lc = 1
rc = 2
lc = 3
rc = 4

Heapify → Maintaining properties of heap after any operation.

```
swap (A, 0, n-1)          n--  // reducing 1 element
i = 0
while (i < n) {
    lc = 2*i + 1
    rc = lc + 1
    if (lc >= n) break      // i → leaf node
    if (rc >= n) {
        if ( A[lc] < A[i] )  swap (A, i, lc)
        break
}
```





```
if ( A[lc] < A[i]  &&  A[lc] <= A[rc]) {
    swap (A, i, lc)
    i = lc
} else if ( A[rc] < A[i]  &&  A[rc] <= A[lc]) {
    swap (A, i, rc)
    i = rc
} else    break
}
```

TC = O(log(N))   SC = O(1)

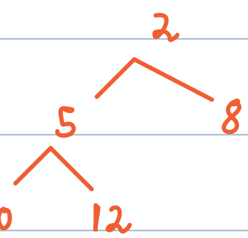
✪ Build a Heap [Interview Problem]

1) Insertion → ∀ elements, insert one by one.

TC = $\underline{O(N \log(N))}$

2) Sorting

	0	1	2	3	4
A =	[2	5	8	10	12]

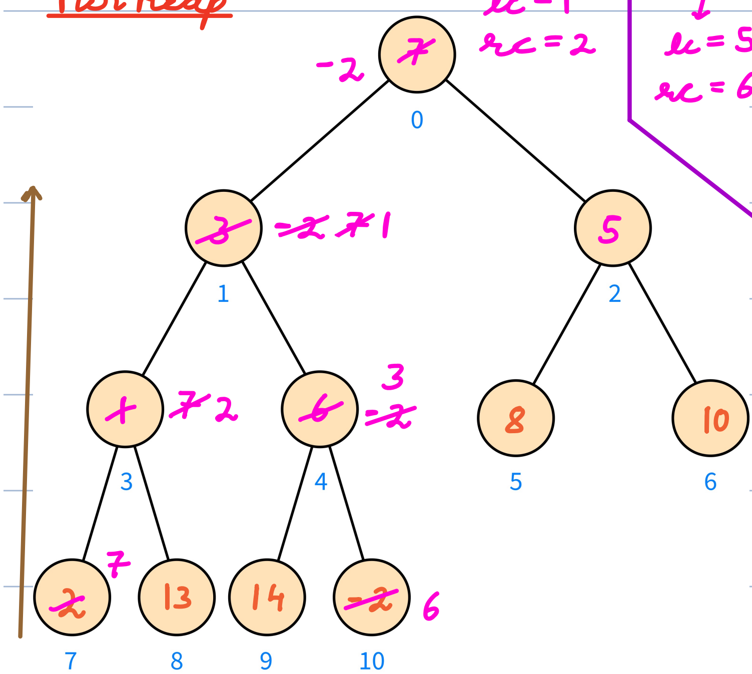


TC = $\underline{O(N \log(N))}$

3) Bottom Up →

A =	[~~7~~ -2	~~3~~ ~~-2~~ ~~7~~ 1	5	~~1~~ ~~7~~ 2	~~6~~ ~~-2~~ 3	8	10	~~2~~ 7	13	14	~~-2~~ 6]
	0	1	2	3	4	5	6	7	8	9	10

Min Heap



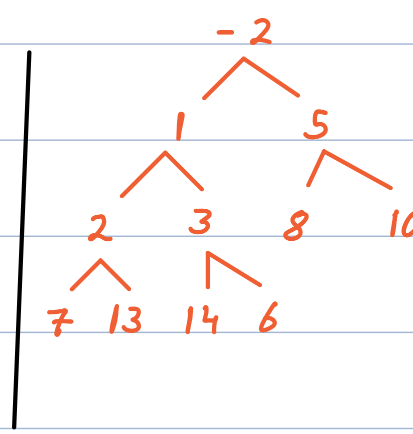
lc = 1
rc = 2

lc = 5
rc = 6

lc = 9
rc = 10

lc = 7
rc = 8

lc = 3
rc = 4

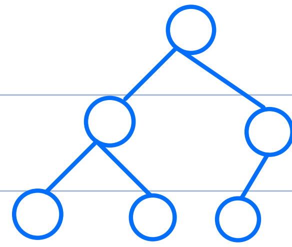


SC = $\underline{O(1)}$

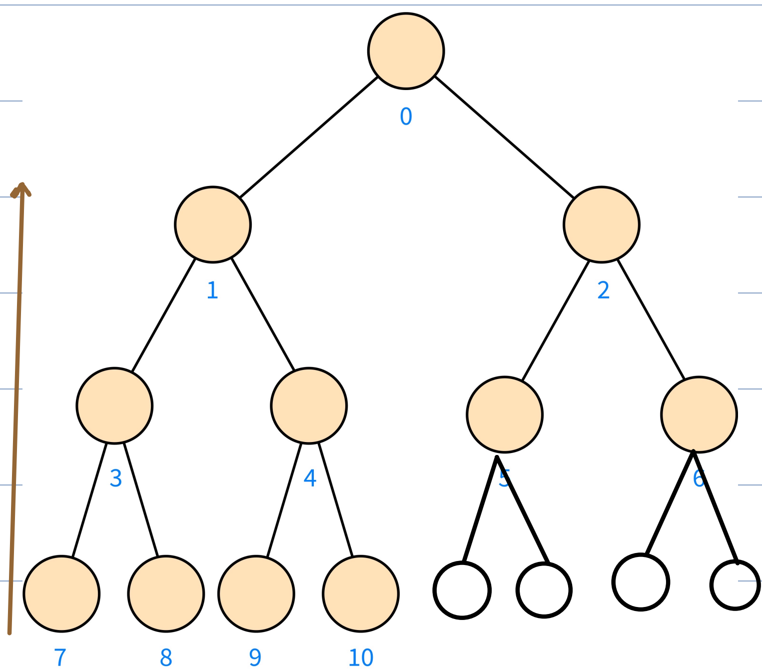
(inplace build)

of leaves in complete binary tree

$N \rightarrow$	1	2	3	4	5	6	...
# leaves $\rightarrow$	1	1	2	2	3	3	...



$\frac{N+1}{2}$ (half)



#nodes	#swaps
N/16	3
N/8	2
N/4	1
N/2	0

$$TC = \cancel{\frac{N}{2} * 0} + \frac{N}{4} * 1 + \frac{N}{8} * 2 + \frac{N}{16} * 3 + \ldots$$

$$= \frac{N}{2} \underbrace{\left(\frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \ldots \right)} = \frac{N}{2} * 2 = N$$

TC = $\underline{O(N)}$

$$S = \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \cdots$$

$$- \quad \frac{S}{2} = \frac{1}{4} + \frac{2}{8} + \frac{3}{16} + \cdots$$

$$\frac{S}{2} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = \frac{1/2}{1 - 1/2} = \frac{1/2}{1/2} = 1$$

$$\Rightarrow S = \underline{2}$$