Q→ 2 5 2 6 3

cost of connecting 2 ropes = sum of length of the ropes.
Find min cost to connect all the rope, connecting two ropes at a time.

2 5 ⇒ cost = 2+5 = 7 +
2 6 ⇒ cost = 2+6 = 8
cost = 7+8 = 15
15 3 ⇒ cost = 15+3 = 18

Total Cost
7
+8
+15
+18
48 ✓

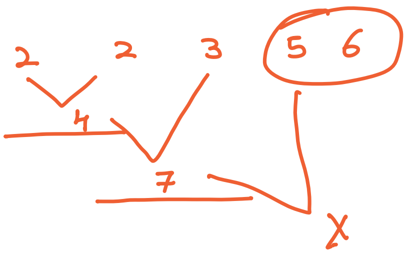
2 + 2 = 4 ✓
4 + 3 = 7
6 5 = 11
7 + 11 = 18

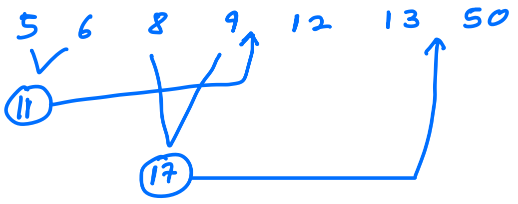
Total Cost = 4
+7
+11
+18
40 ✓

$x < y < z$

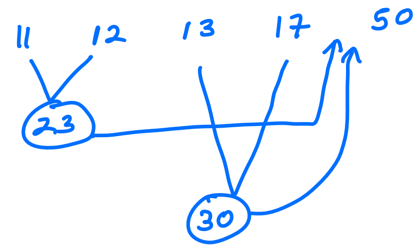
$$\underbrace{(x+y) + (x+y)+z}_{\text{Case 1}} < \underbrace{(x+z) + (x+z)+y}_{\text{Case 2}} < \underbrace{(y+z) + (y+z)+x}_{\text{Case 3}}$$

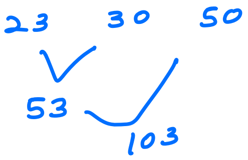
Cost = 4 + 7





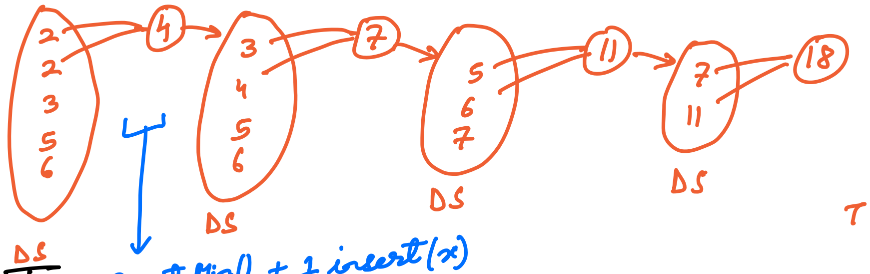
Cost = 11 + 17 + 23 + 30
+ 53 + 103 ✓





Insertion Sort → $TC = O(N^2)$

DS → insert(x) ✓ } $O(\log(N))$ ✓
→ getMin() }



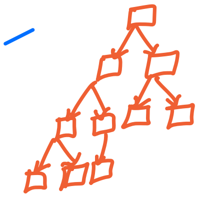
2 getMin() + 1 insert(x)

Total Cost
4 + 7 + 11 + 18 = 40

$TC = O(N\log(N))$

DS → Binary Heap

1) Structure → complete Binary Tree ✓



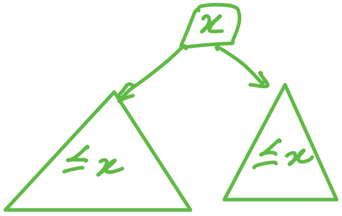
2) Types of Heaps

MinHeap

✓ ∀node node.data ≤ node.left.data
node.data ≤ node.right.data

MaxHeap

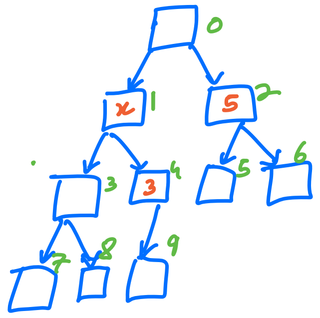
∀nodes node.data ≥ node.left.data
node.data ≥ node.right.data



✓ 3) No relation b/w left & right subtree.

Heapify → Properties of heap should be maintained after any operation (insert/delete).

Array Representation of tree (level order traversal)



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

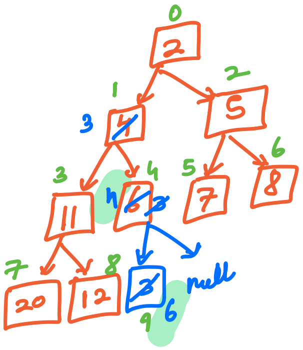
✓

$i \to$ left child $= 2*i+1$

$i \to$ right child $= 2*i+2$

$i \to$ parent $= (i-1)/2$

Insert



Min Heap

→

0	1	2	3	4	5	6	7	8	9
2	~~4~~	5	11	~~6~~	7	8	20	12	~~3~~
↑	3 ↑			~~3~~ 4					6

✓

Insert(3)

$TC = O(\log(N))$

$i = 9 \to \text{parent} = \frac{9-1}{2} = 4$

$A[4] > A[9] \Rightarrow \text{swap}(4,9)$

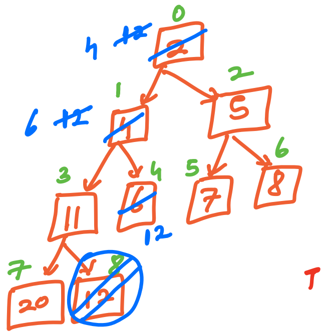
$i = 4 \to \text{parent} = \frac{4-1}{2} = 1$

$A[1] > A[4] \Rightarrow \text{swap}(1,4)$

$i = 1 \to \text{parent} = \frac{1-1}{2} = 0$

$A[0] < A[1] \Rightarrow$ **stop** ✓

Get Min



Min Heap

→

0	1	2	3	4	5	6	7	8	9
~~2~~	~~4~~	5	11	~~6~~	7	8	20	~~12~~	
~~12~~ 4	~~12~~ 6			12 ↑					

getMin()

$TC = O(\log(N))$

$i=0 \to lc = A[1] = 4$ min(4, 5) = 4

$i=0 \to rc = A[2] = 5$

$i=1 \to lc = A[3] = 11$ min(11, 6) = 6

$i=1 \to rc = A[4] = 6$

$i = 4 \to lc = A[9]$, $rc = A[10]$] empty ✓✓

Delete / Update / search → $TC = O(N)$
(Any random location)

Build Heap 1) Insert all elements 1 by 1 ⇒ $TC \approx O(N\log(N))$ ✓

$TC = \log(1) + \log(2) + \log(3) + \dots \log(N)$

$= \log(1*2*3*\dots N) = \log(N!)$

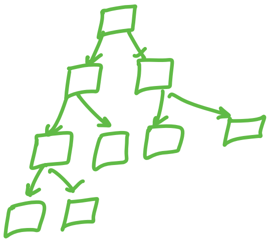
$\boxed{\log(a)+\log(b) = \log(a*b)}$

2) If all elements are known, can we do in better TC ?

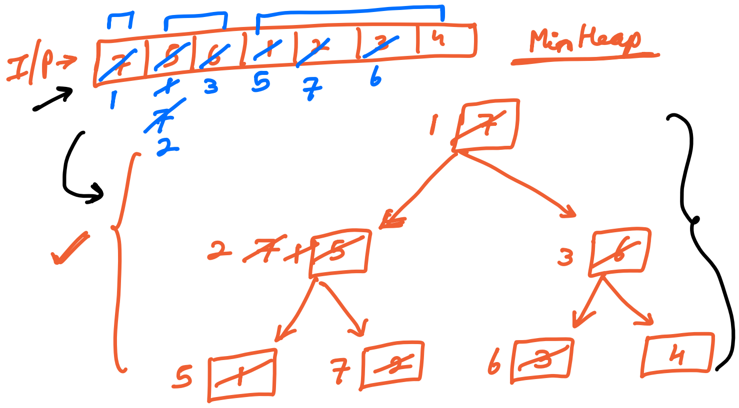
#leaf nodes in complete binary tree, if total nodes = N → $\frac{(N+1)}{2}$ ✓

→ 1 , 1 , 2 , 2 , 3 , 3 , 4 , 4 , 5

→ 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9



$\frac{7+1}{2} = 4$ leaf nodes



$\simeq N/8$ swaps = 2 per node

$\simeq N/4$ swaps = 1 per node

$\simeq \frac{N}{2}$ swaps = 0

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

$= \frac{N}{2}\left(\frac{1}{2}+\frac{2}{4}+\frac{3}{8}+\dots\right) = \frac{N}{2}\boxed{\sum_{i=1}^{\infty}\frac{i}{2^i}}$ ✓ $= \frac{N}{2}*2 = N$

$TC = O(N)$ ✓

subtract

$S = \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{4}{16} + \dots$

$\frac{S}{2} = \frac{1}{4} + \frac{2}{8} + \frac{3}{16} + \dots$

$\frac{S}{2} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \dots \Rightarrow \frac{S}{2} = \frac{1/2}{1-1/2} = \frac{1/2}{1/2} = 1$

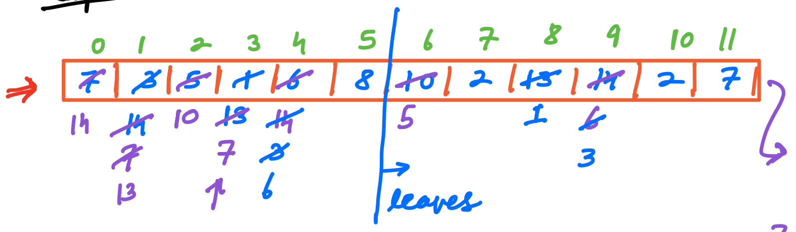
$\Rightarrow \boxed{S=2}$

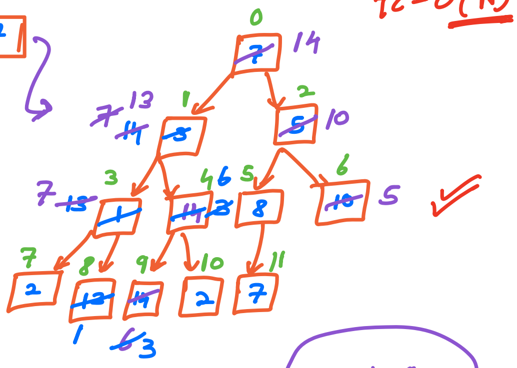
Inplace Heap Build

Build a Max Heap ✓

$SC=O(N) \Rightarrow O(1)$ ✓

$TC=O(N)$





i = 5 → lc = A[11] = 7
→ rc = A[12] = X

A[5] > A[11] ✓

i = 2 → lc = A[5] = 8
→ rc = A[6] = 10 ✓

i = 4 → lc = A[9] = 14 ✓
→ rc = A[10] = 2

i = 3 → lc = A[7] = 2
→ rc = A[8] = 13 ✓

i = 1 → lc = A[3] = 13
→ rc = A[4] = 14 ✓
→ lc = A[9] = 6 ✓
→ rc = A[10] = 2

i = 0 → lc = A[1] = 14 ✓
→ rc = A[2] = 10
→ lc = A[3] = 13 ✓
→ rc = A[4] = 6
→ lc = A[7] = 2 ✓
→ rc = A[8] = 1

Q→ N chocolate bags, each having A[i] chocolates.

Kid → selects a bag with highest # chocolates & eats it.

Magician → Fill the bag again with → A[i]/2 chocolates.

Find # chocolates kid can eat in K steps.

A → [~~10~~, 3, ~~15~~, ~~8~~, 4]
~~5~~ ~~7~~ 4
2 3

K = 5

15 + 10 + 8 + 7 + 5 = 45

Sol → 1) construct max heap → $TC=O(N)$ $SC=O(1)$

2) get Max() for every step & insert (max/2) → $TC=O(K \log(N))$

$TC=O(K \log(N)+N)$ $SC=O(1)$