Mohammed Varaliya

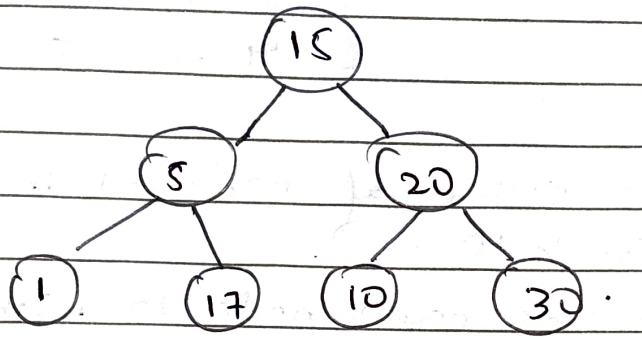
Roll No: 54

Assignment question.

Apply heapify method to build a heap for the following list.

[15, 5, 20, 1, 17, 10, 30].

Soln:



15	5	20	1	17	10	30
1	2	3	4	5	6	7

To find the leaf node

$$\text{formula} = \left(\frac{n}{2} + 1 \text{ upto } n\right)$$

$n = \text{len}(arr) = 7$.

$$= \left(\left\lfloor\frac{7}{2}\right\rfloor + 1 \text{ upto } 7\right)$$

$$= (3 + 1 \text{ upto } 7)$$

$$= (4 \text{ upto } 7)$$

= | 1 | 17 | 10 | 30 | this all are the leaf node.

will try to make a max heap from the node

leaf[start] - 1.

= 4 - 1

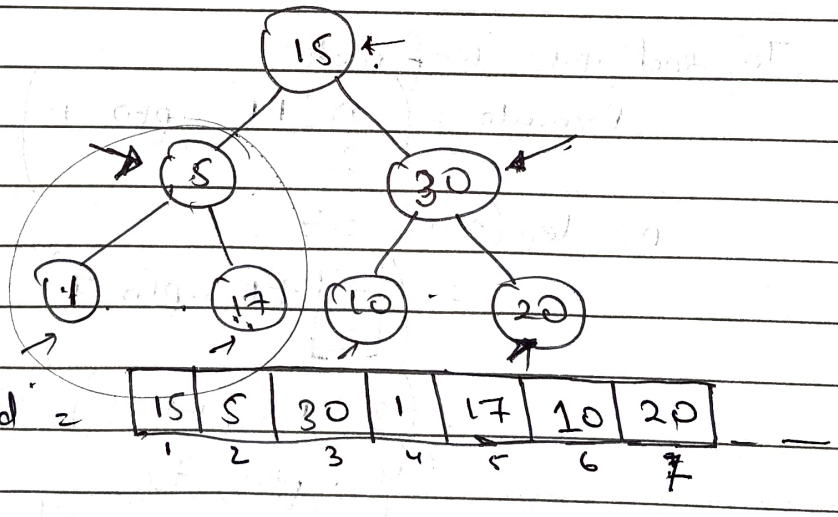
= 3.

Step 1: find left and right child of Node at index 3 → [20]

↳ left child = 2 * i
= 2 * 3
= 6 → [10]

↳ Right child = 2 * i + 1
= 2 * 3 + 1
= 6 + 1
= 7 → [30]

compare siblings of node at index i → 3 → [20]
∵ 30 > 10
∴ compare parent node 20 with child 30
∵ 30 > 20.
∴ interchange.



Step 2: decrement i by 1 i.e, i = 2
find left and right child of node at index 2 → [5]

↳ left child = 2 * i
= 2 * 2
= 4 → [1]

15	05	30	1	17	10	20

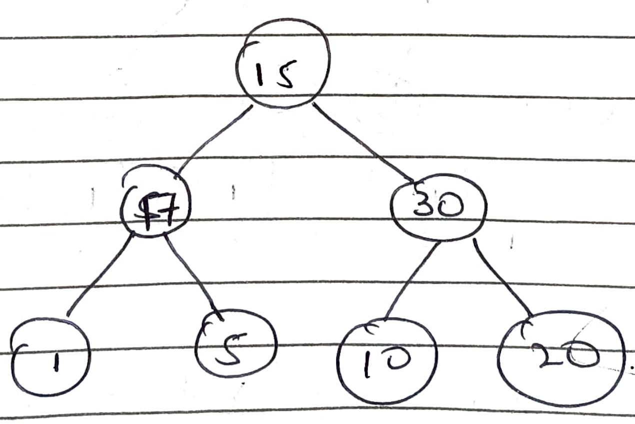
↑→
1

Krantee Ma'am's work

7
2*7 = 14 } Termination
2*7+1 = 15 } cond.

↳ right child = $2*i+1$
= $2*2+1$
= $4+1$
= 5 → [17]

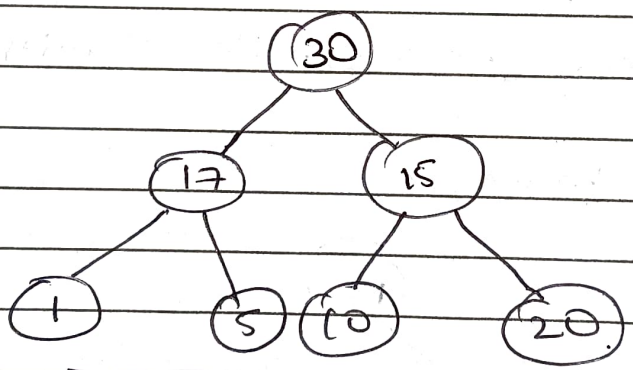
compare siblings of node at index i = 2 → [5]
∵ 17 > 1
∴ compare parent node 5 with child 17.
∵ 17 > 5
∴ interchange.



updated list =

15	17	30	1	5	10	20
1	2	3	4	5	6	7

Step 3: decrement i by 1 i.e i = 1.
find left and right child of node at index 1 → 15
↳ left child = 2 * i
= 2 * 1
= 2 → [17]
↳ right child = 2 * i + 1
= 2 * 1 + 1
= 2 + 1
= 3 → [30]
Compare siblings of node at index i → 1 → [15]
∵ 30 > 17
∴ compare parent node 15 with child ~~17~~ 30.
∵ 30 > 15
∴ Interchange.



Updated List =

30	17	15	1	5	10	20
1	2	3	4	5	6	7

Step 4: Now, when we reached to the very first pIndex of the heap, will again iterate from start to end to make sure our heap is in a max heap format.

increment i ~~with~~ by 1, i.e. i = 2.

find left and right child of node at index 2 → [17]

↳ left child = $2 * i$

$= 2 * 2$

$= 4 \longrightarrow [1]$

↳ Right child = $2 * i + 1$

$= 2 * 2 + 1$

$= 4 + 1$

$= 5 \longrightarrow [5]$

compare siblings of node at index $i = 1 \rightarrow [17]$

∵ $5 > 1$

∴ compare parent node 17 with child 5

∵ $17 > 5$

∴ Nothing change, keep as it is

Step 5: increment i by 1, i.e $i = 3$

find left and right child of node at index $3 \rightarrow [15]$

↳ left child = $2 * i$

$= 2 * 3$

$= 6 \longrightarrow [10]$

↳ right child = $2 * i + 1$.

$= 2 * 3 + 1$

$= 6 + 1$

$= 7 \longrightarrow [20]$

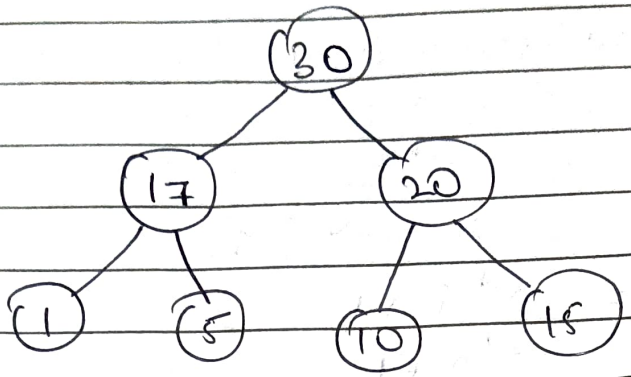
compare siblings of node at index $i = 3 \rightarrow [15]$

∵ $20 > 10$.

∴ compare parent node 15 with child 20.

∵ $20 > 15$.

∴ Interchange.



updated list =

30	17	20	1	5	10	15
1	2	3	4	5	6	7

Step 6: increment i by 1, i.e i = 4 → [1]

find left and right child of node at index 4 → [1]

↳ left child = 2 * i

= 2 * 4

= 8 → [index out of bounce]

↳ right child = 2 * i + 1

= 2 * 4 + 1

= 8 + 1

= 9 → [index out of bounce]

will immediately terminate the loop, and return the current list/heap, as we are pretty sure, that it is max heap.

hence, the final heap/list returned by func is

30	17	20	1	5	10	15
1	2	3	4	5	6	7