

Heap

9.00 Heap is a type of DS which exist
9.30 in form of

10.00 Complete Binary Tree
10.30 +
11.00 And that holds heap property
11.30

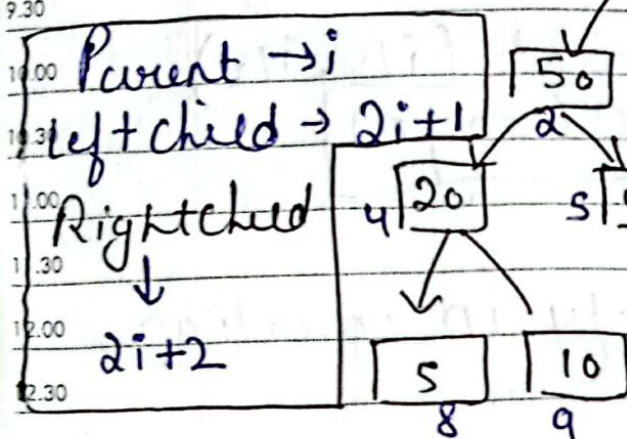
12.00 → Heap → DS
12.30

1.00 CBT +
1.30 That holds heap
2.00 property.
2.30

3.00 CBT
3.30 → all levels are completely filled
4.00 (except last level)
4.30
5.00 → Filling element from
5.30 (L → R)
6.00

Implement Heap Array

0 based Indexing



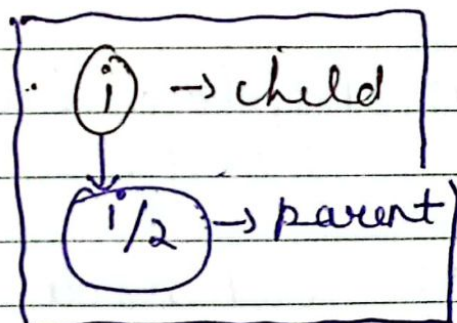
1 based Indexing
Parent $\rightarrow i$

Left child $\rightarrow 2i$
Right child $\rightarrow 2i+1$

-1	100	50	60	20	40	45	55	5	10
0	1	2	3	4	5	6	7	8	9

1 based Indexing

Child to parent



Insertion

- 1) add node at the end of array
- 2) swap if greater than its parent.
(max heap over min heap)

Insertion Time Complexity

arr[size-1] = 57 $\rightarrow O(1)$

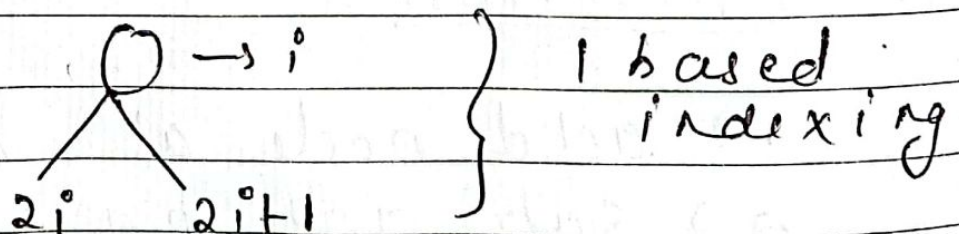
&
then compare it parent $(i/2, i/2)$
which make $O(\log n)$

Total Time Complexity in insertion
 $\rightarrow O(\log n)$

Deletion \rightarrow In Heap, we only
can delete element
from Root.

1) Replace Last value with root
node.

2) Root node ke correct position
pr bhejna.



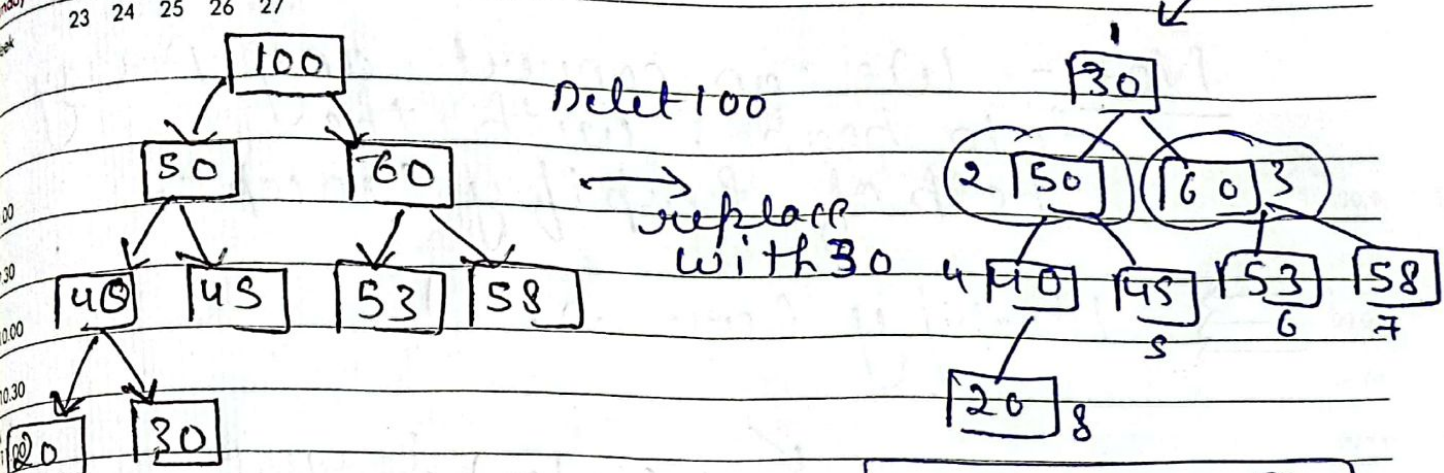
	6	13	20	27	J
	7	14	21	28	N
1	8	15	22	29	U
2	9	16	23	30	N
3	10	17	24	-	2
4	11	18	25	-	0
5	12	19	26	-	2
23	24	25	26	27	

Max Heap

MAY 2022

wednesday
(124-241) Week 19

4



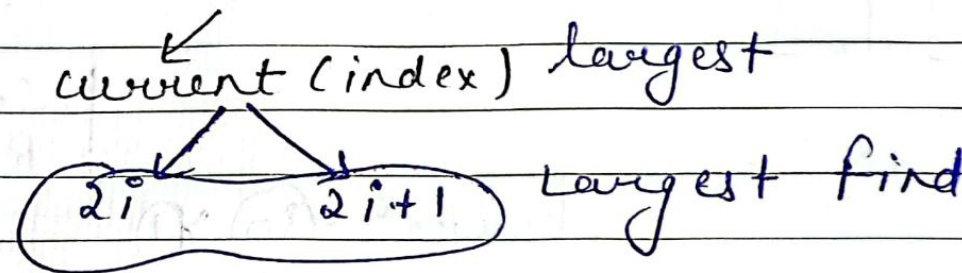
left = 50 → ②
 Right = 60 → ③
 largest = ③ → ①
 largest = ③
 compare → 60

Time Complexity

- 1) $O(1)$ → Make comparison at level
- ② $O(\log n)$ → levels + k comparison krega.

Note - We can convert any Array into heap, with the help of heapify concept.

⇒ Heapify (arr, n, i)



$i == \text{largest}$
at correct
position

$i \neq \text{largest}$ (Need to swap
and update i with
largest index)

Heapify Time Complexity
→ $O(\log n)$

Wednesday	2	9	16	23	30	2
Thursday	3	10	17	24	-	0
Friday	4	11	18	25	-	2
Saturday	5	12	19	26	-	2
Sunday	23	24	25	26	27	

friday
(126-239) Week 19

6

Build heap from Array

No need to check for leaf node

→ $\left(\frac{n}{2} + 1\right)$ to n → all are leaf Node.

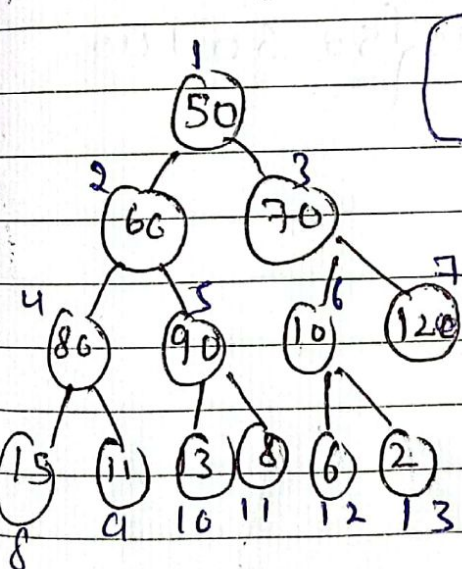
In complete Binary tree

0 based indexing

1 based indexing

Parent → i
Left → $2i + 1$
Right → $2i + 2$

Parent → i
Left → $2i$
Right → $2i + 1$



heapify first

10
↓
40
↓
80
↓
12

10 → 90 → 80 → 70 → 60

$\frac{13}{2} + 1$ to n

7 to 13

leaf node

leaf Node
 $\left(\frac{n}{2} + 1\right)$ to n
No heapify
 $\frac{n}{2} > 0$ (Do heapify)

JUNE

JULY

AUGUST

MAY 2022

7

saturday

(127-238) Week 19

I + work like
selection sort

M	Monday	30	2	9	16
Y	Tuesday	31	3	10	17
	Wednesday		4	11	18
2	Thursday		5	12	19
0	Friday		6	13	20
2	Saturday		7	14	21
2	Sunday	1	8	15	22
Week		18/23	19	20	21

Heap sort

- 9.00 ① swap first & last element
9.30 ② heapify
10.00

100 50 60 20 30

heapify →

[30 50 60 20] {100}

60 50 30 20 {100}

Heapify →

20 50 30 {60 100}

50 20 30 {60 100}

Sort

30 20 {50 60 100}