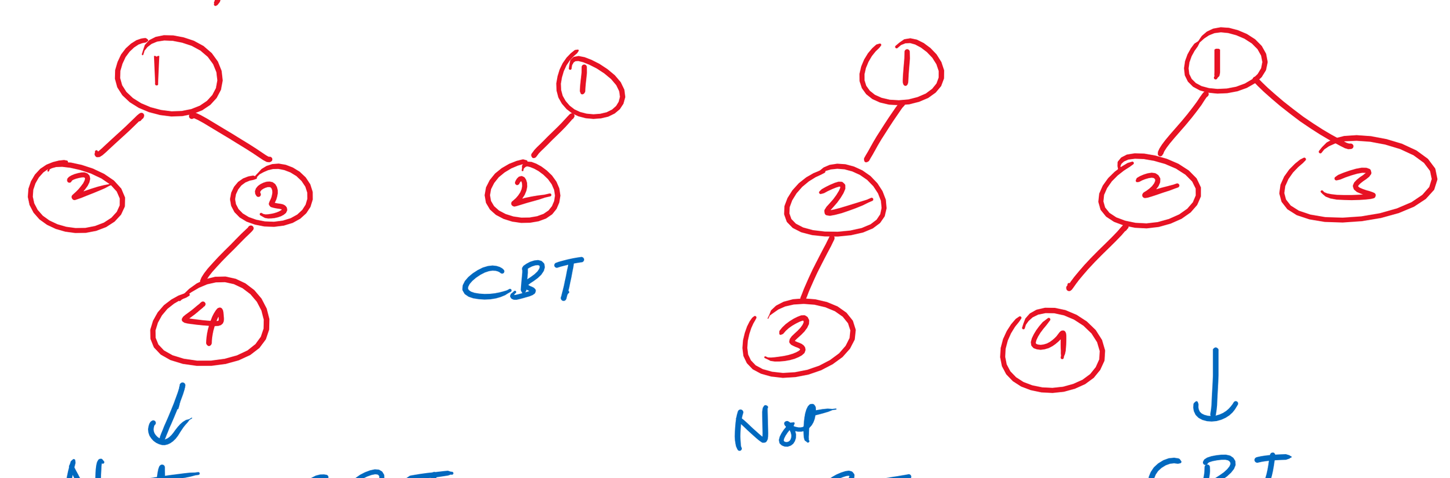
* Complete Binary Tree

Heap Data Structure



A CBT is a tree where insertions are done Top to Bottom & Left to Right.

The Heap Data Structure: → heap

Normally in a BST, the time complexity of finding/searching an element in the BST is Big O of log (N).

But, during program execution, Someone comes & tells you to find the max or min value in Constant Time O(1), then we use the Heap concept.

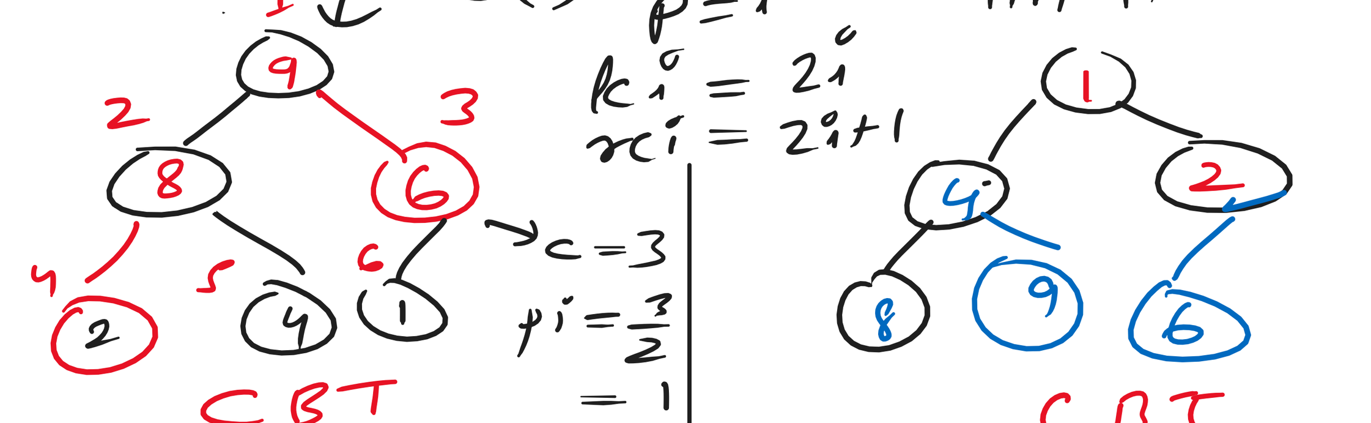
Based on this logic, we have two types of heaps:
(i) Max Heap (ii) Min heap

heapify

arr[] = 2, 8, 6, 4, 9, 1

Mx H O(1) Mn H

$p = i$
$lci = 2i$
$rci = 2i+1$



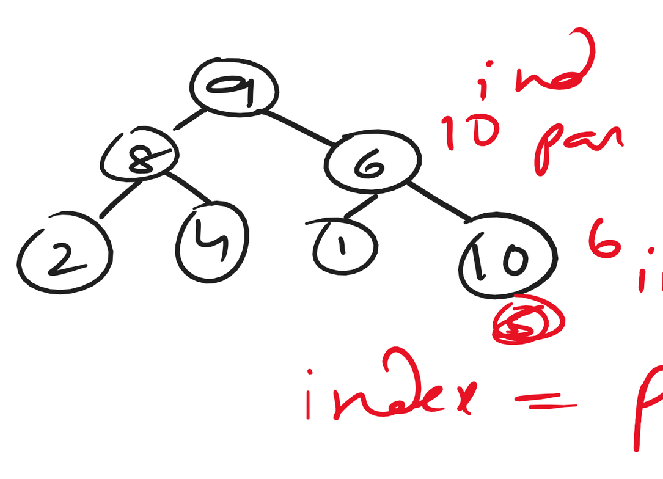
-1	9	8	6	2	4	1
0	1	2	3	4	5	6

7

↓ sentinel no

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

↳ sentinel number



-1	9	8	6	2	4	1	10
0	1	2	3	4	5	6	7

Size++ = 7

index = par arr[size] = val = 10;

p = 3/2 = 1

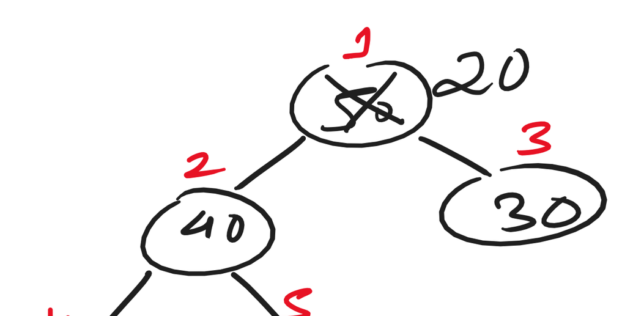
index = val;
parent = 7/2

20, 30, 50, 10, 40

50	40	30	10	20

if arr(index)>

swap (i,p) arr[parent)



Max Heap

-1	50	40	30	10	~~20~~
0	1	2	3	4	5

Size

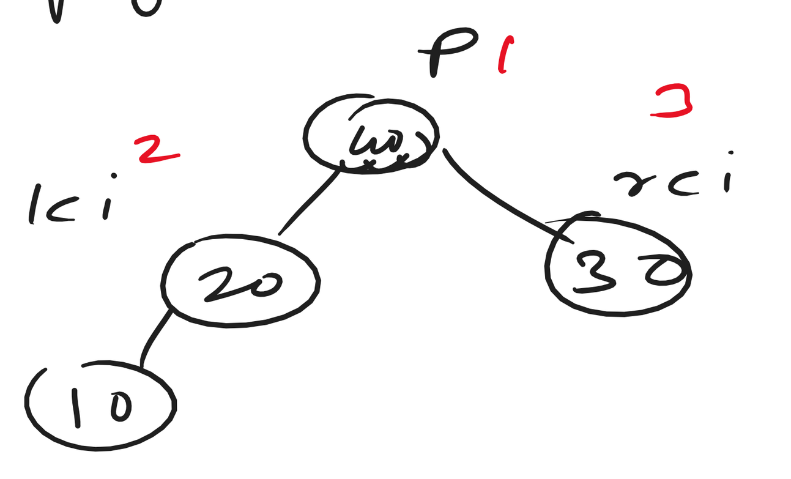
Delete the largest → root (50)

arr[1] = arr[size]

arr[1] = 20

size --;

Propagate root to its correct pos



completed

40	20	30	10

if arr[lci] > arr[p] swap

else arr (rci] > arr(p) swap

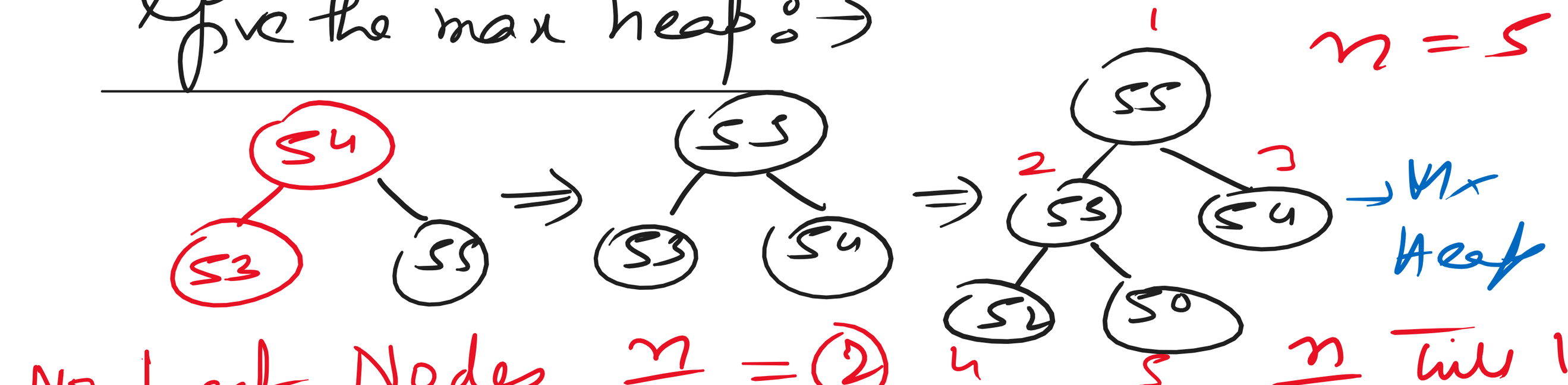
The process of converting an array into a max or min heap is called "heapify"

Consider: arr = {-1, 54, 53, 55, 52, 50}
n = 5

Give the max heap: →

55	53	54	52	50

n = 5



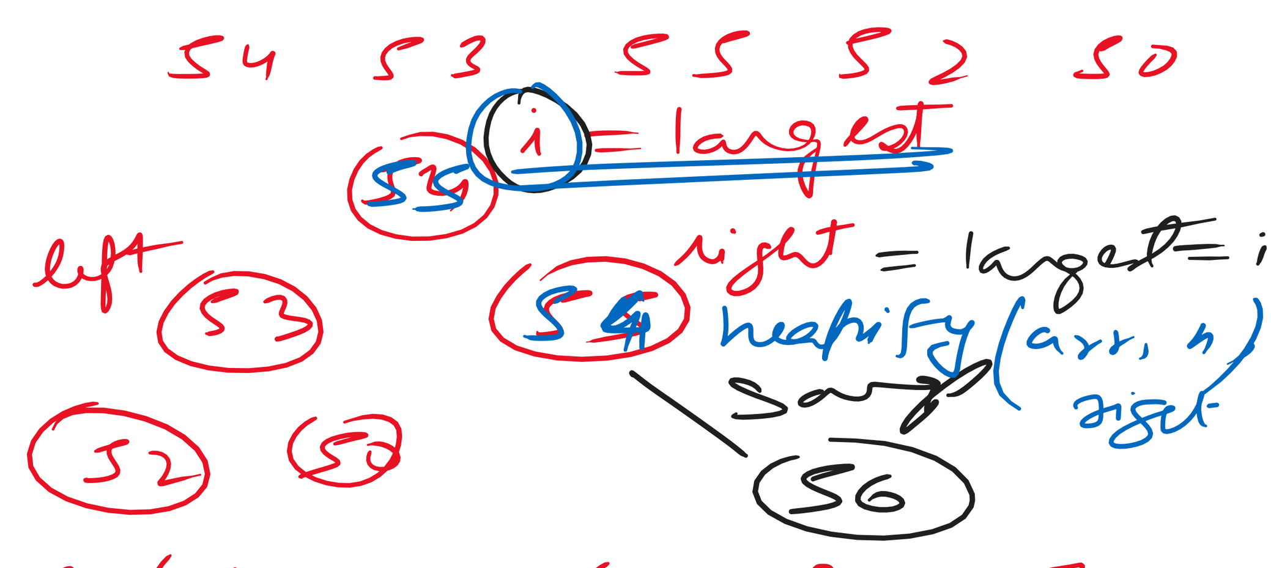
Non-Leaf Nodes $\frac{n}{2} = 2$ $\frac{n}{2}$ till 1

$\frac{n}{2} + 1 = 2+1 = 3$ Leaf nodes

Non-leaf nodes $(0 <= x \leq \frac{n}{2})$

(55, 53, 54, 52, 50)

54 53 55 52 50



if (left <= n && arr[largest] < arr[left])
largest = left;

if (right <= n && arr[largest] < arr[right])
→ largest = right;

Heap Sort:

n × log n
Ascending order

-1	55	53	54	52	50
0	1	2	3	4	5

size

Swap arr[1] & arr[size]

54 | 53 | 50 | 52 | ~~55~~ size--

↳ correct pos → heapify(1)

52 | 53 | 50 | ~~54~~ size--

53 52 50

55	53	54	52	50
1	2	3	4	5

size

not a heap

swap(arr[1], arr[size])

50	53	54	52	55

↳ heapify (a, n, 1) sorted size--

54 53 50 | 52 ~~55~~

52 53 50 | 54 → sorted

50 52 | 53 54 55