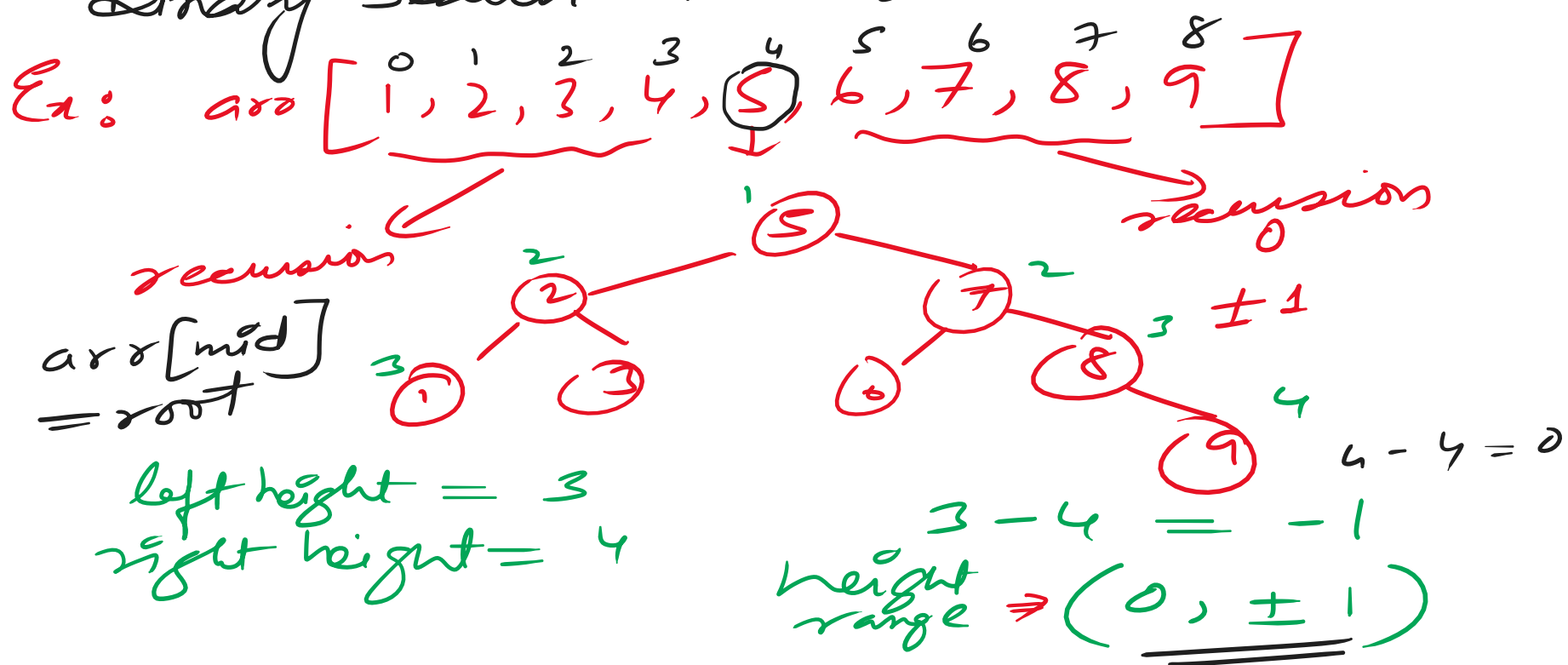
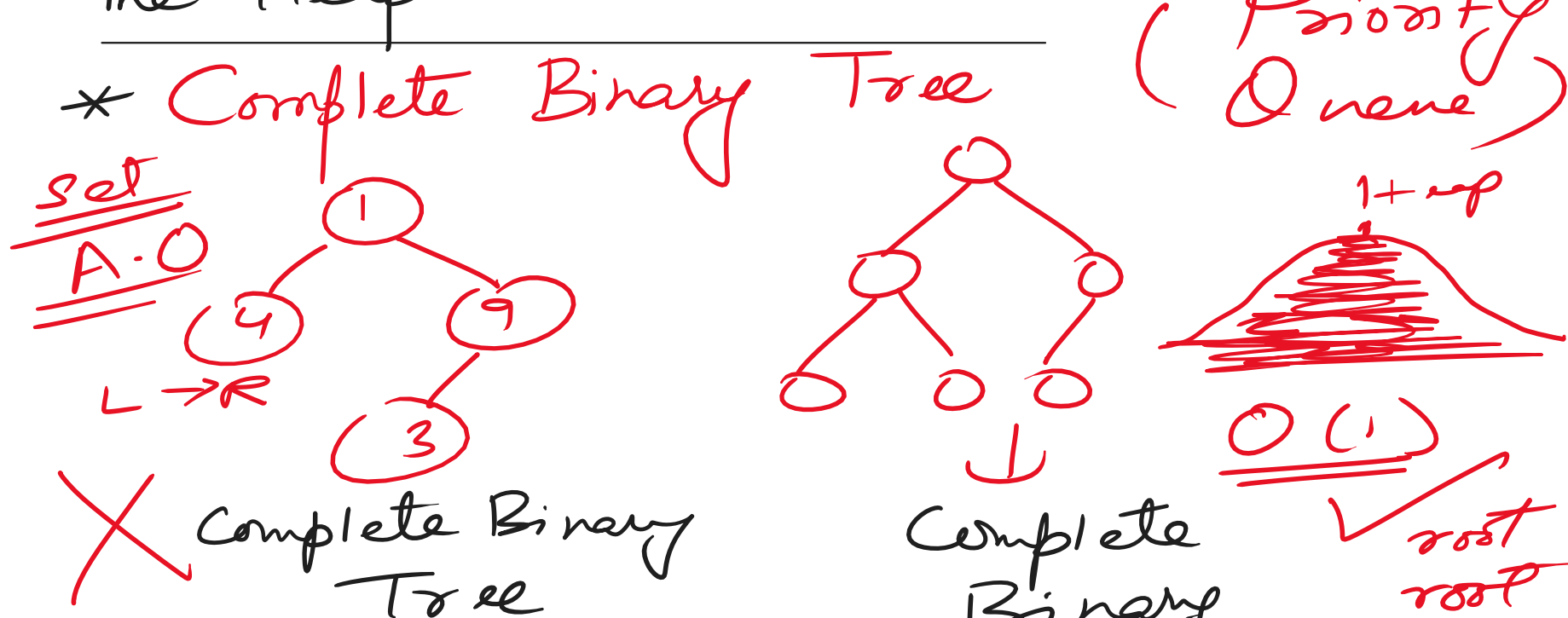


* Convert a given sorted array to a Binary Search Tree :



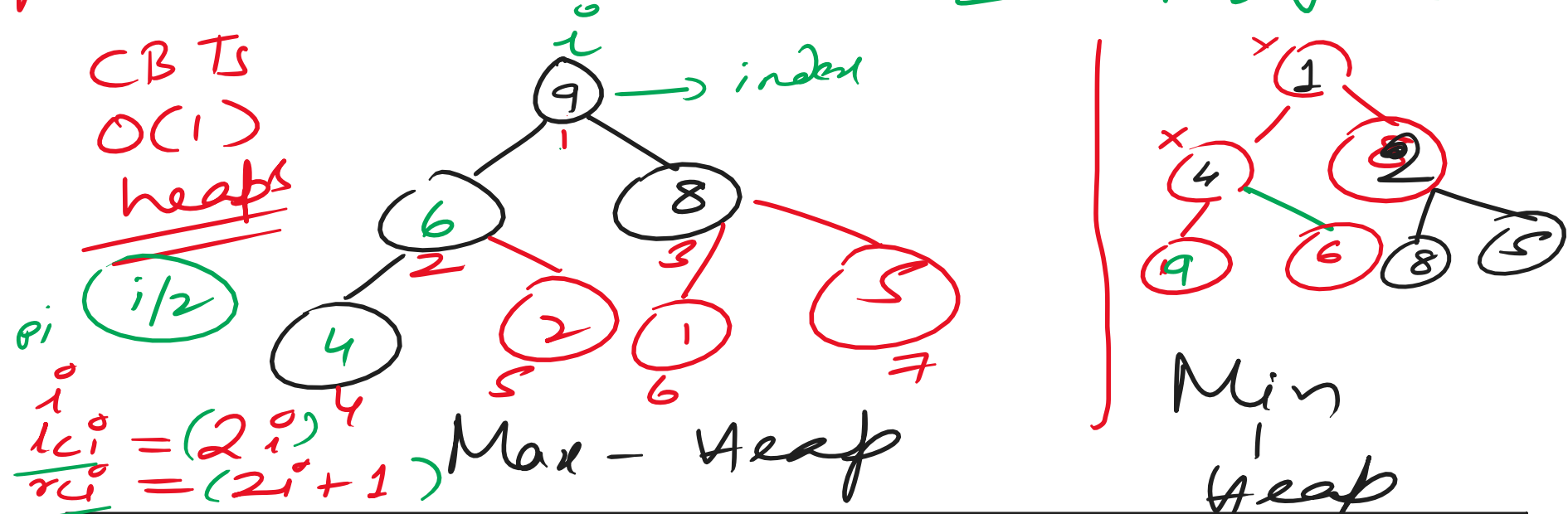
The Heap Data Structure :->



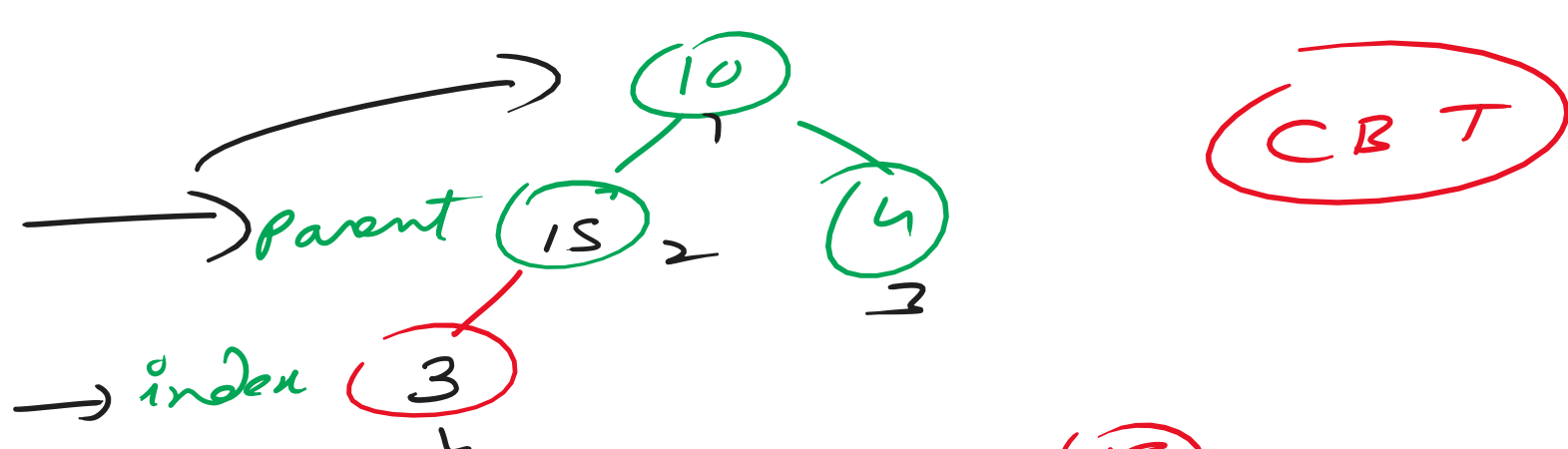
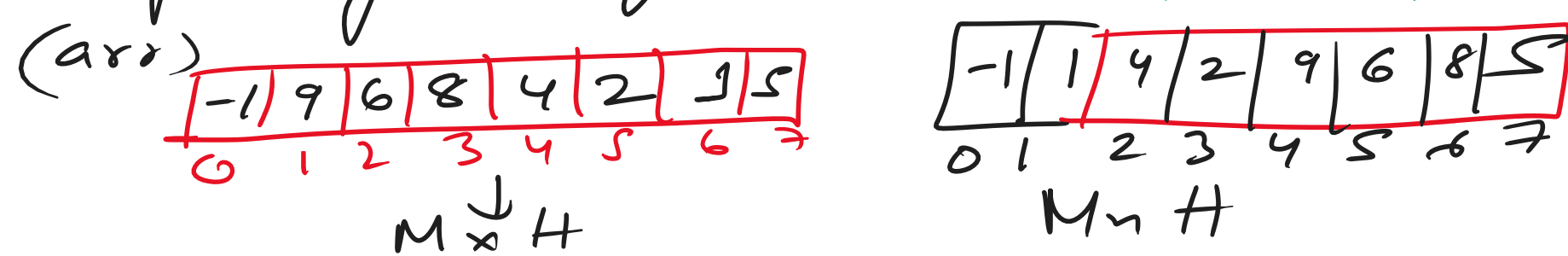
Insertions -> Leaf nodes as well.

Max-Heaps -> [4, 9, 8, 6, 2, 1, 5]

$P_i = i/2$ arr -> heap [heapify]



Greedy Algos (K largest, K smallest)



$arr[parent] < arr[index]$

Swap (p, i);

index = parent;

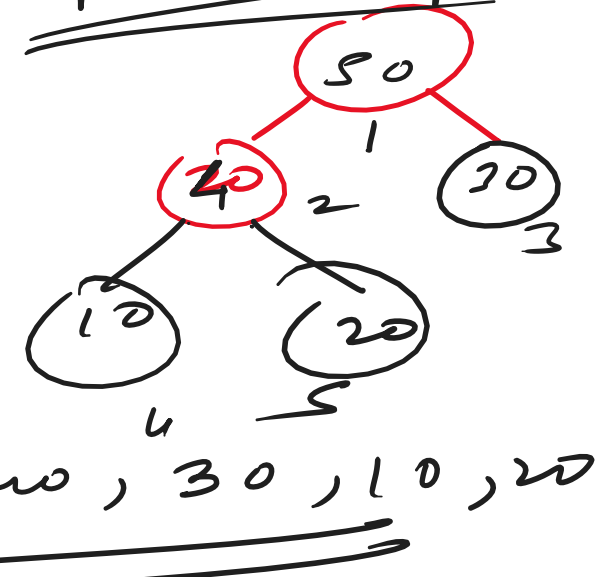
2 -> 1

[20, 30, 50, 10, 40]

insert(20)
30
50
10
40

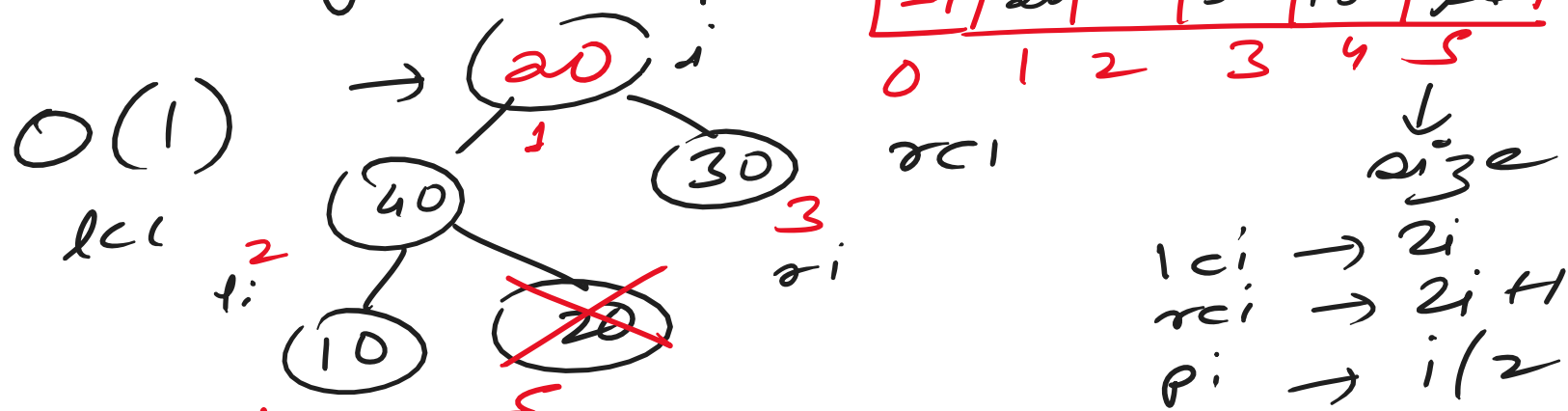
->

Max Heap



[50, 40, 30, 10, 20]

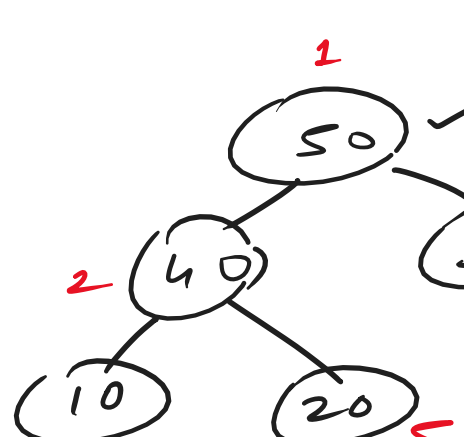
* Delete from heap :->



① $arr[1] = arr[size]$

② $size--$

③ Propagate the root to its correct position.



$n = 5$
no. of nodes

* No of leaf nodes in a heap -> $\frac{n}{2} + 1$

(Non-leaf nodes) = $\frac{5}{2} + 1 = 2 + 1 = 3$

Heapify -> arr -> heap

*** While converting array into heap, we only use heap property for the non-leaf nodes.

[Heapify / Heap Sort / Problems]

** Tries

R, G

* Back Tracking

* Greedy Algos -> (8)

* Graphs

* Dynamic Programming -

* Bit Manipulations.

(Ceil / Floor) -> BST

bigdatatraining.com

Session Code : 14178