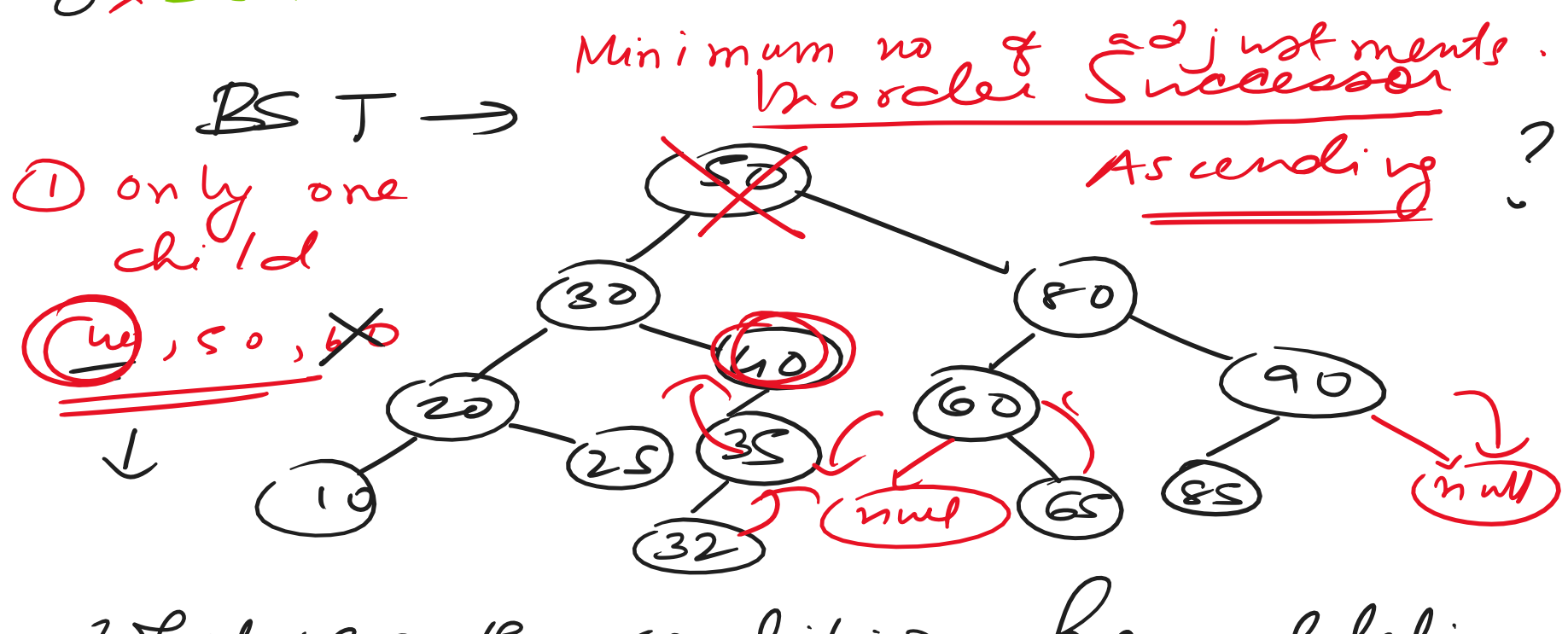
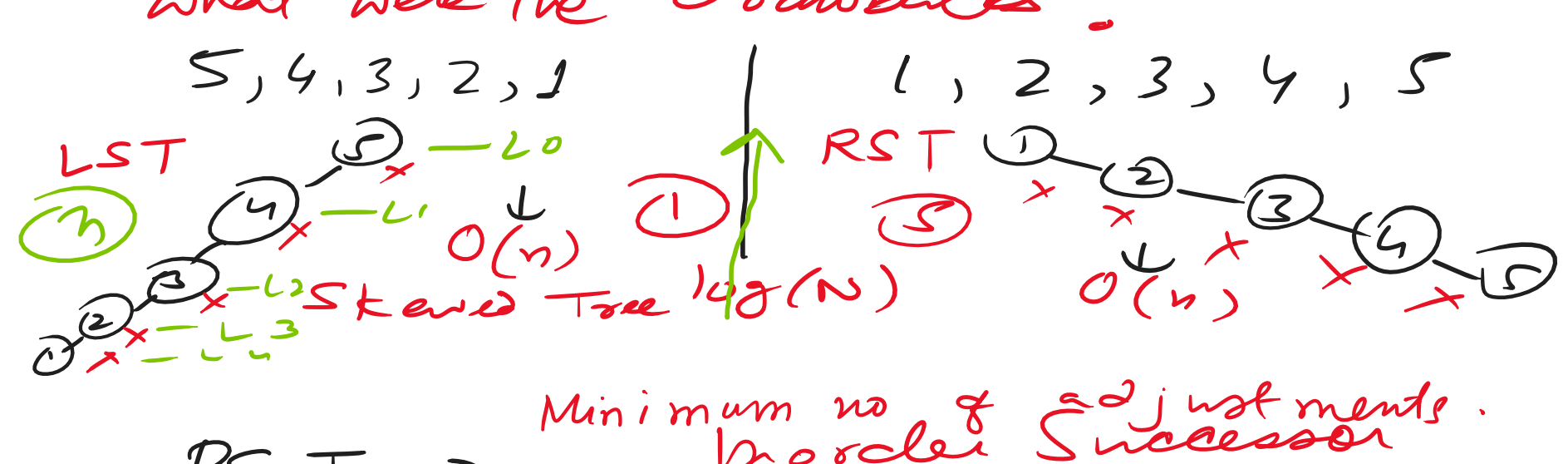


- \* Arrays
- \* Searching & Sorting
- \* Linear Data Structures
  - ↳ stacks
  - ↳ queues
  - ↳ linked lists
- \* Non-linear data structure
  - ↳ Trees → Binary Trees
  - ↳ Rotations → Binary Search Trees
  - [AVL, Red Black] → Heaps (CBT)

BST → Strictly height balanced.  
 What were the drawbacks?



What were the conditions for deleting a node in a BST?

Already, Prim's  
 ✓ Max Heap

Parent > Children  
 PE = i  
 LCP = 2i  
 RCP = 2i+1

heapify

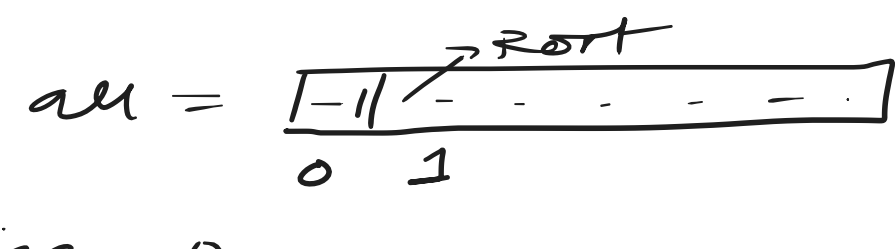
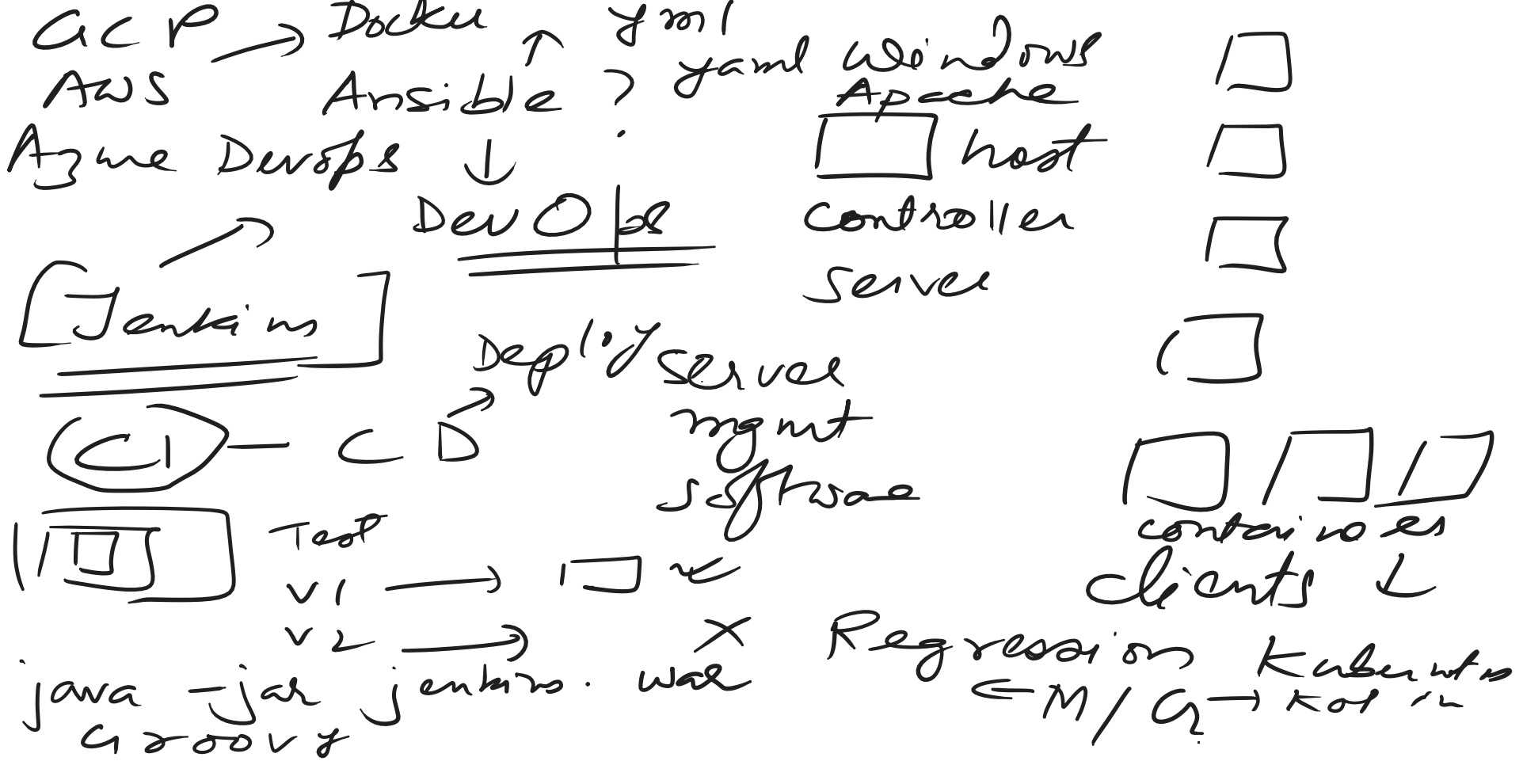
Why heaps?

Arrays (Priority queue)  
 CBT  
 Min Heap

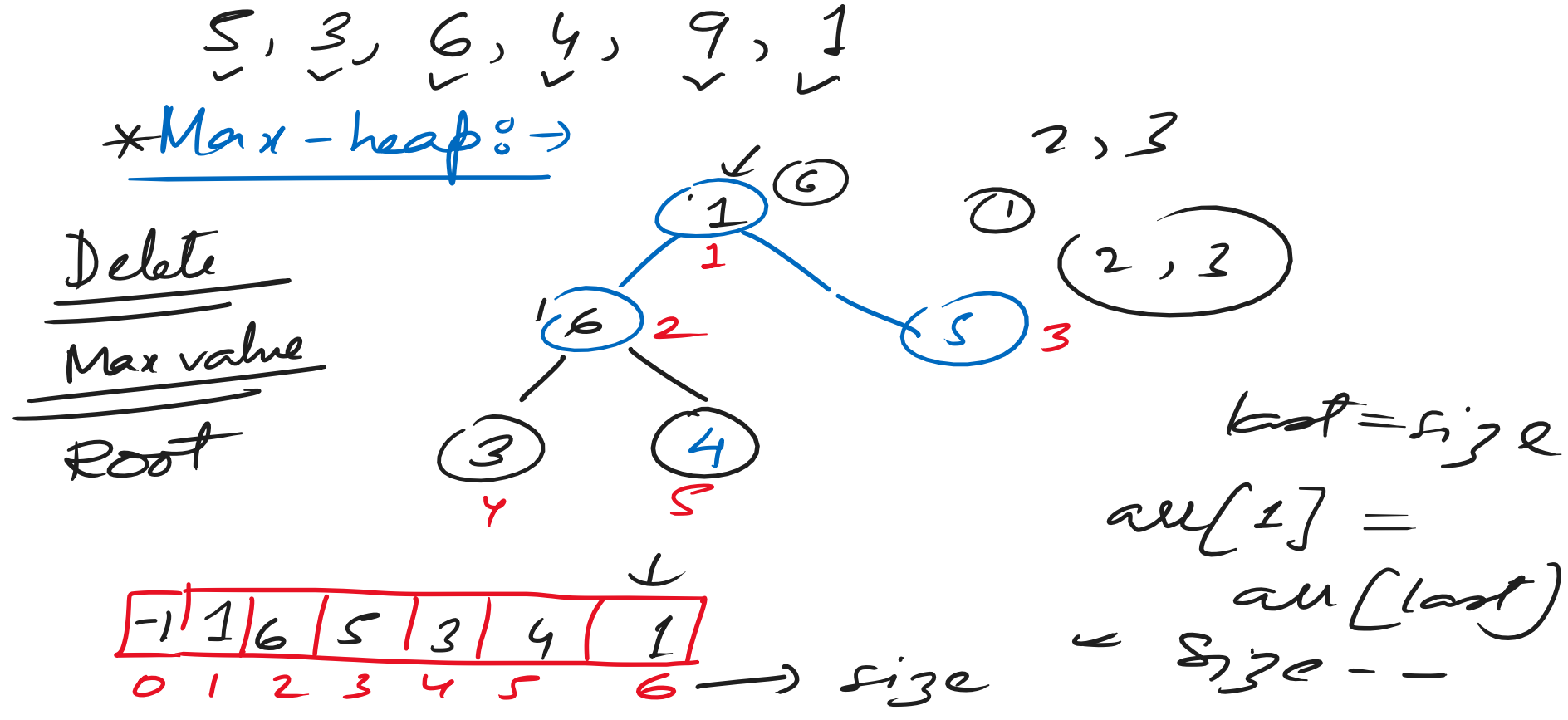
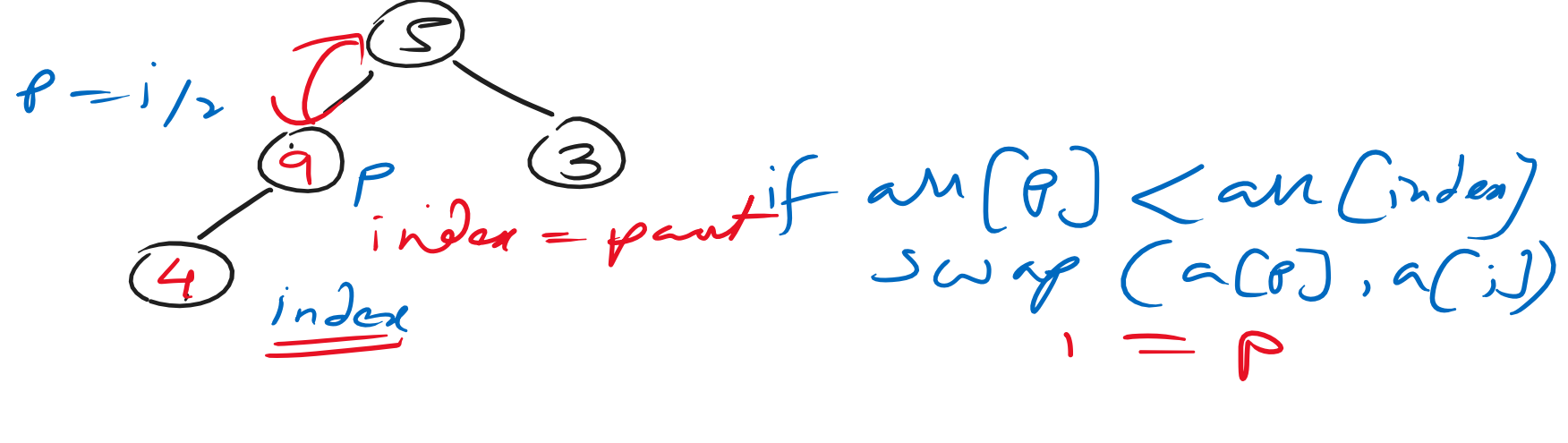
Parent < Children  
 PE = i  
 LCP = 2i  
 RCP = 2i+1

BFS  
 Queue

Why heaps?

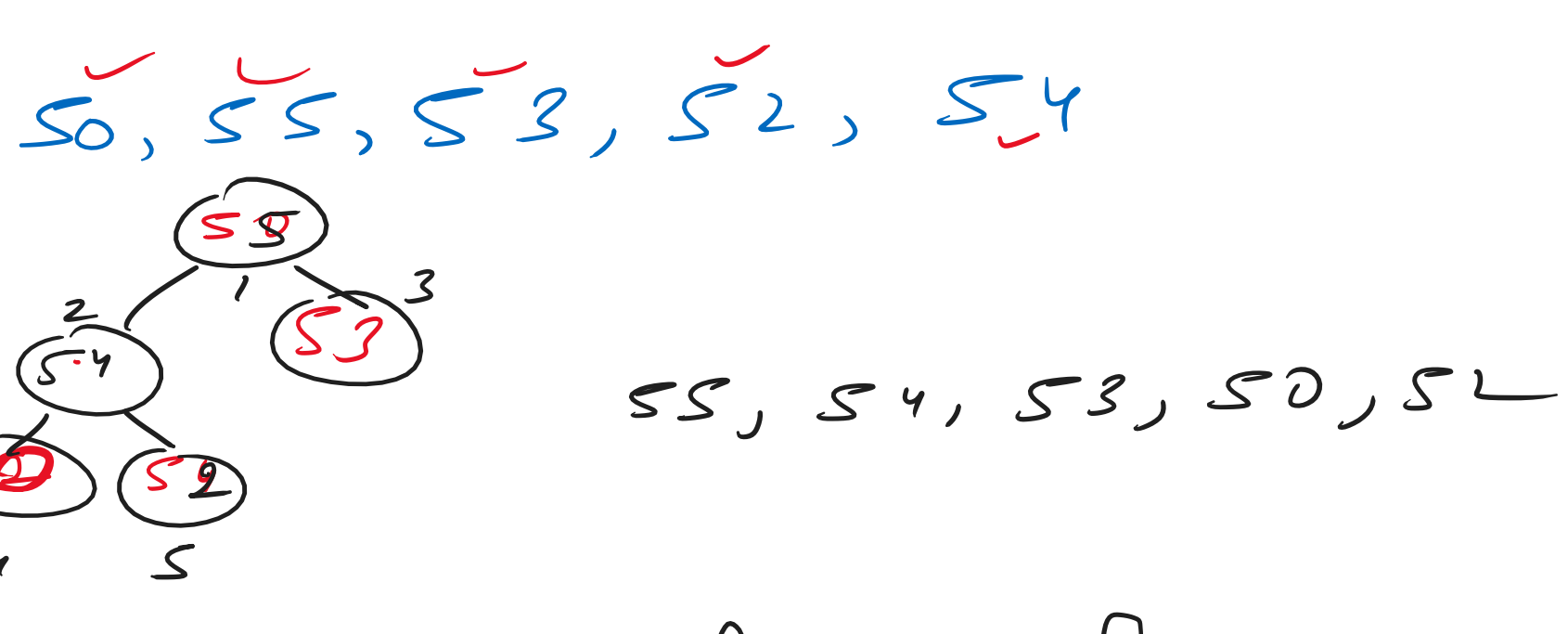


Size = 0  
 Size -- size + 1  
 Size ++



arr → heap  
 → heapify  
 → heap sort  
 heap → asc  
 → desc

maxchild element = max(arr[lci], arr[rcl])  
 if arr[i] < mce



Heapify: → The process of converting an array into a heap is called heapify.  
 (n/2 till 1) (Non leaf nodes)  
 arr = { -1, 54, 53, 55, 52, 50 }  
 largest i = 55  
 left 2: 53  
 right 2+1: 54  
 recursion: 52, 50  
 Important point: → How many leaf nodes are present in a heap?  
 n = 5 nodes  
 leaf nodes = n/2 + 1 = 2 + 1 = 3

