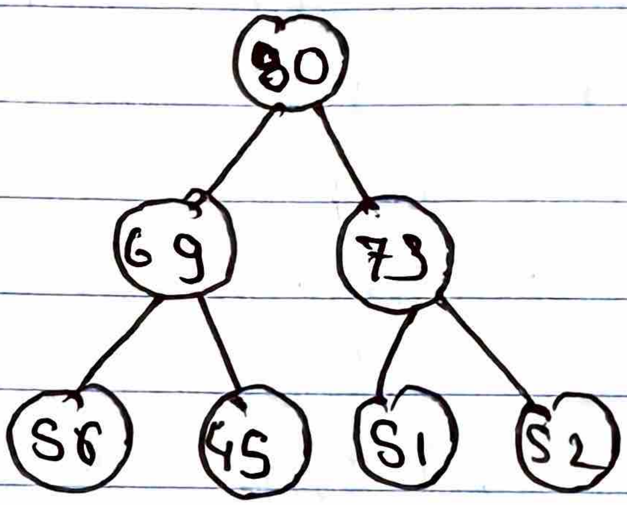
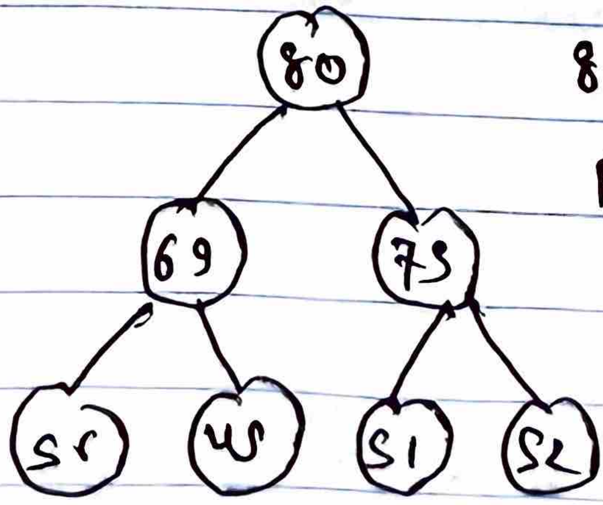
* **Heaps :-** In general we talk about binary heap where a node can have maximum of 2 child



There are 2 types of heap
i> max heap
ii> min heap

i> **Max heap** :- In this to parent or root node must be greater than it's child and same follows for the sub-tree.



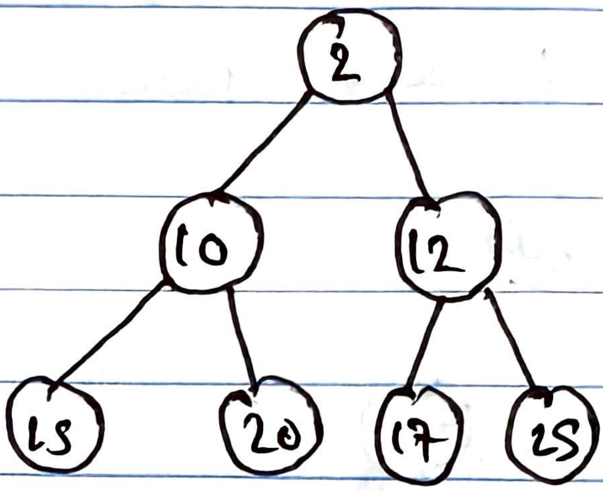
80 > 69, 73

For sub-tree.

69 > 56 and 45

75 > 51 and 52

Min heap :- In this the root node must be smaller than the child node.



2 < 10 & 12.
10 < 15 & 20
12 < 17 & 25

* height of any tree can be calculated using $\boxed{h = \log(n)}$

* When adding items into array it has to be followed level wise and left to right.

0	1	2	3	4	5	6
80	69	73	56	45	51	52

As this is structured tree so the ~~ind~~ i for left $\boxed{2i+1}$ and right $\boxed{2i+2}$. To get parent Node:

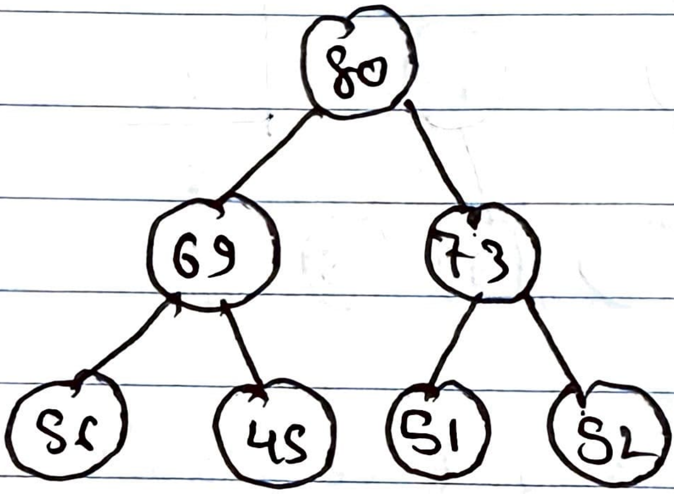
$i = \frac{i}{2} - 1 \Rightarrow \frac{4}{2} - \frac{1}{1} \Rightarrow \boxed{1}$

69 is Parent

* Search and Heap together is a bad situation,

* Search has O(n) complexity.

* Inserting a number which is great than root node.



80	69	73	56	45	51	52

* Suppose adding a number 83.

* First simply add at the end.

80	69	73	56	45	51	52	83
0	1	2	3	4	5	6	7

* Let calculate the parent of 83

$7/2 - 1 = 5/2 \approx 2.5$

upper part of 2.5 is 3 $\boxed{36}$ is the parent. Let's swap the position. If 85 > 56 then swap.

80	69	73	83	45	51	52	56

Again same step :-

80	85	73	69	45	51	52	56

again :-

85	85	73	69	45	51	52	56

→ Insert complexity log(n). or O(log n). In best case O(1)

* Deletion :- If deleting last elemen O(1), If working with Binary heap we are only allowed to delete root node. O(log n).

* Heap Sort ! Heap sort means sort either acending to descending or descending to ascending.

* In terms of max heap we'll go from min to max.

* **Priority Queue :-** In priority Queue the FIFO is not followed the element with priority and normal treated seperately. Priority Queue is implemented by heap.

* Heapify is a process of converting any binary tree into a compliment heap.