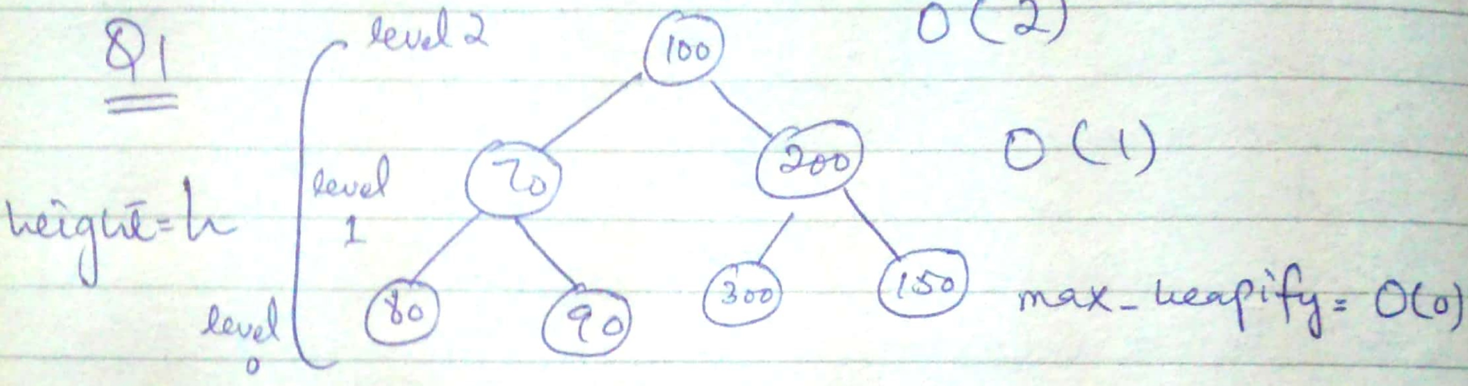
Momina Atif Dar

P18-0030

Q1



Because the leaves are already heaps so we don't heapify them. And as the levels go up from leaves to root node, time complexity to heapify increases.

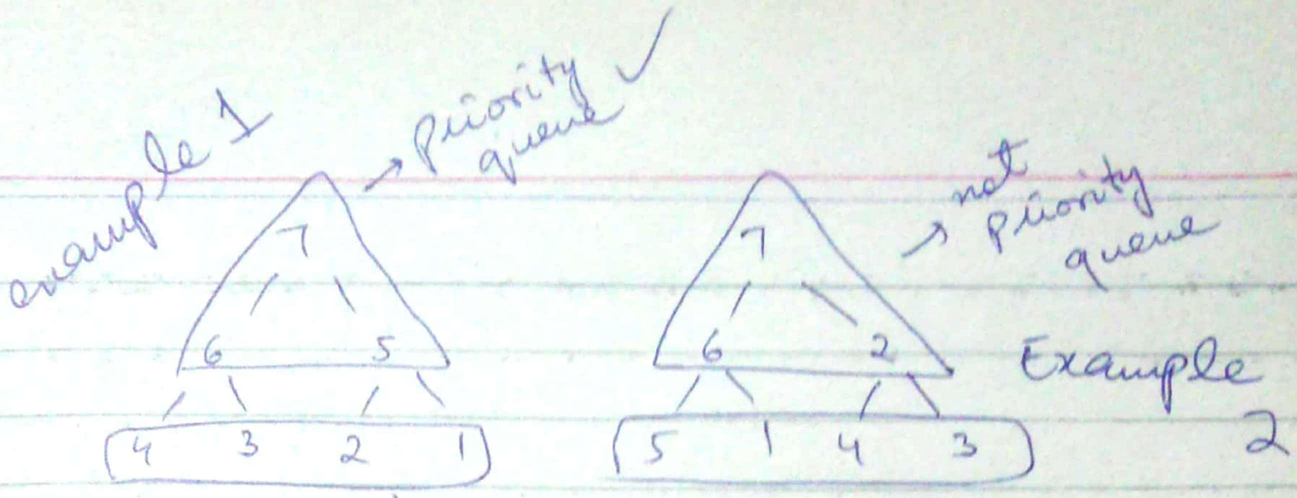
Example: 200 will be heapified and swapped with 300. As 200 is on height 1 so it would take O(1) time.

Then, 300 and 100 would be swapped which will take O(2) time because 100 is at height 2.

So, running time of max_heapify would be O(h) because it depends on height.

$$O(h) \downarrow O(\log n).$$

Q2



~~We if we don't ignore leaves then~~

To check priority queue we have to check parent nodes. If the parent node is max at every level then it's priority queue.
So, if we check all nodes then running time would be $O(n)$ but it's not asymptotically tight.

We know leaves are ~~already~~ already heaps so running time would be less than $O(n)$.
In above mentioned example 2 we have to check only 7, 6 and 2.

check $\lfloor \frac{n}{2} \rfloor$ nodes.

nodes at each level : $\lceil \frac{n}{2^{h+1}} \rceil$

to check if it's max : $O(1)$

So, running time = $O(1) \times \lceil \frac{n}{2^{h+1}} \rceil$