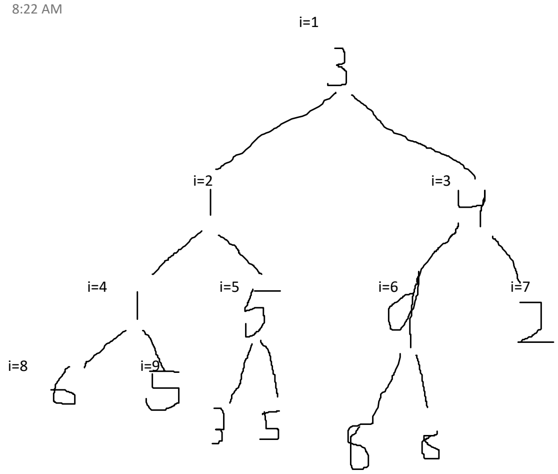
Monday, October 25, 2021 8:22 AM



Complete - every level (Except possible last) is full. And last fills in from left.

Become a heap if every parent >= Child

The reason we haven't talked about the reason this needs to be a complete binary tree is because you can store it in an array - tree structure is implicit

If parent index is i, children indicies are 2i and 2i+1
Paren of node I are x = floor(i/2)

Heap Sort is O(nlgn)

Build a heap:
    Bottom up o(n)

First finds the index of the largest element that still has children.
In this case the index would be 6 with element 9. 9 is greater than all of its children so it's good

Now work backwards, check the 5 node. 5 is >= than all of its children so it's good

1 is not good so 6 and 1 are switched

Keep doing this

Repeat:
    Remove Largest element - o(1)
    Reform tree - o(lgn)