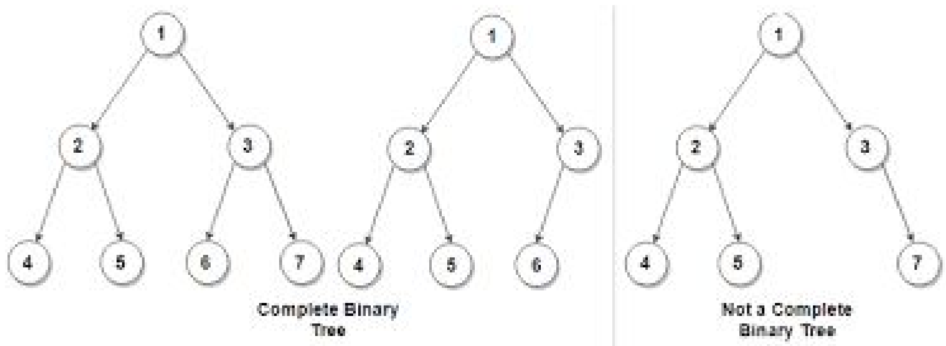
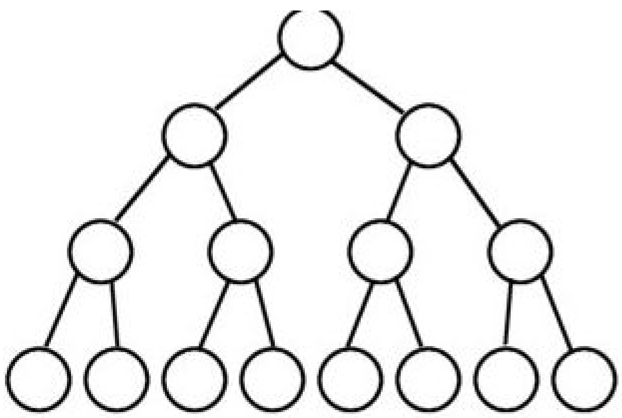
HEAPS

Date: 12-06-2020

1. Heap is a complete binary tree. By complete means: If you do a level order traversal, there's no level, where there is a missing element in the previous level.
   Ex:



2. A full binary tree is a tree in which every node other than the leaves has two children.
   Ex:



3. A heap can be represented by an array with the following logic, for a node at ith index, assuming indexing starts from 1.
   a. 2*i indicates the left child of node i
   b. 2*i+1 indicates the right child of node i
   c. i/2 indicates the parent of node i

4. Heaps are of two types:
   a. Max heap (parent is greater than children)
   b. Min heap (parent is smaller than children)

5. Complexity:
   a. Building heap: may seem O(N logN), but amortized complexity is actually linear.
   b. Insert element: O(log N)

    c. Delete element: O(log N)

6. Applications:
    a. Heap sort: Building heap is O(N), and deleting N-1 elements, total complexity is O(N logN). Every time the first element is replaced with the last element. Size of array decreases by 1. Heapify process is done from the 1st element. Add the end the original array becomes a sorted array
    b. Priority queue

7. The first element in the array always represents the root of the heap.

8. To remove/delete the min/max element
    a. Replace the last element of the array with the first element.
    b. Decrease the length of the array by 1.
    c. Apply heapify process from the root

9. Insert element:
    a. Add the element as the last element in the array.
    b. Recursively keep swapping the element with it’s parent until it’s less than or greater than it’s parent for a max heap or min heap respectively.

10. Heaps to be used whenever we have so many queries involving finding the max element or min element.