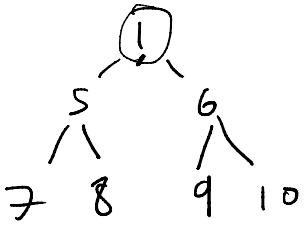
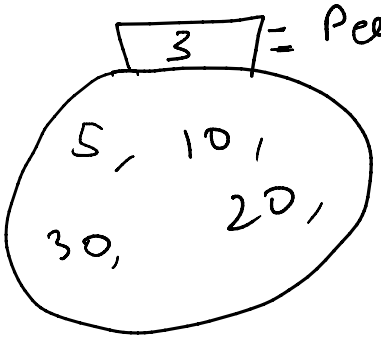
30-Dec-2021

30 December 2021 20:58

Priority Queue :-





min PQ

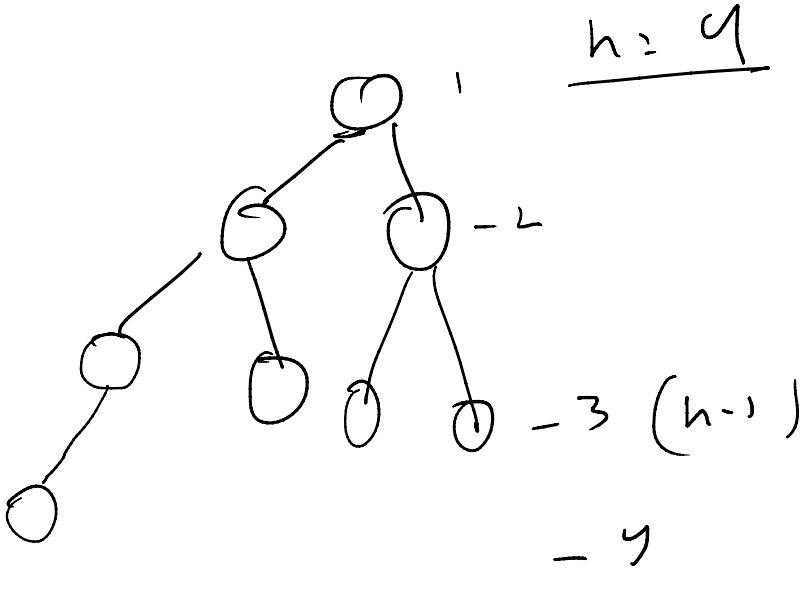
max PQ

	TC
add -	log (n) ✓
remove -	log (n) ✓
Peek -	O(1) ✓
isEmpty -	O(1) ✓
size() -	O(1)

Properties:-

- Parent should always be either largest or smallest
- It should be a complete binary tree till (h-1)

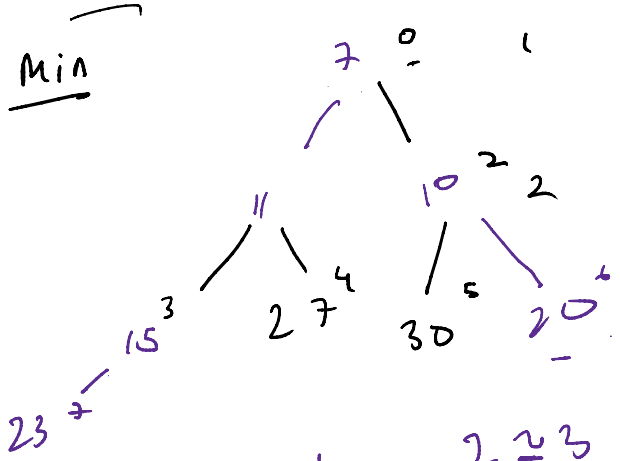
h = height of the tree



0 1 2 3 4 5 6 7

al = 10, 15, 20, 23, 27, 30, 7, 11

Min



$l_i = 2i + 1$ | $P_i = \frac{(i-1)}{2}$

$r_i = 2i + 2$

$2 \times 0 + 1 = 1$

$2 \times 0 + 2 = 2$

$2 \times 1 + 1 = 3$

$2 \times 1 + 2 = 4$

$P_6 = \frac{6-1}{2} = 2$

$\frac{7-1}{2} = 3$

2 ≥ 3

up heapify

n = 7

$2^3 = 8 = 171$

$2^k = n$

$k = \log_2(n)$

$2 \geq 3 = \log(n)$

$P_4 = \frac{4-1}{2} = 1.5 = 1$

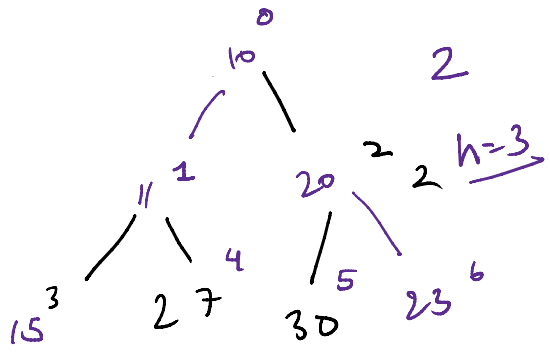
$P_1 = \frac{1-1}{2} = 0$

$2^k = n$

$k = \log_2(n)$

$2i + 1$

$2i + 2$



Peek = root

down heapify

T.C = log (n)

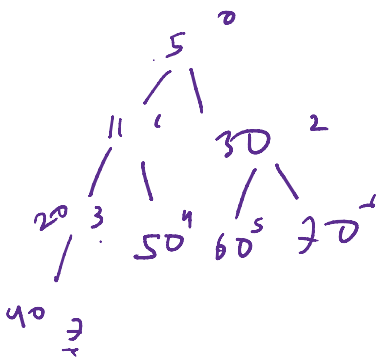
PQ {

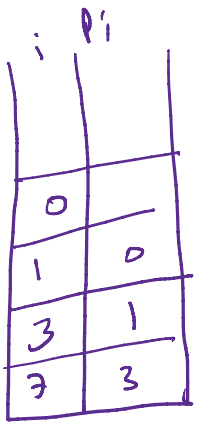
ArrayList<>

ArrayList

{

upheapify() =)

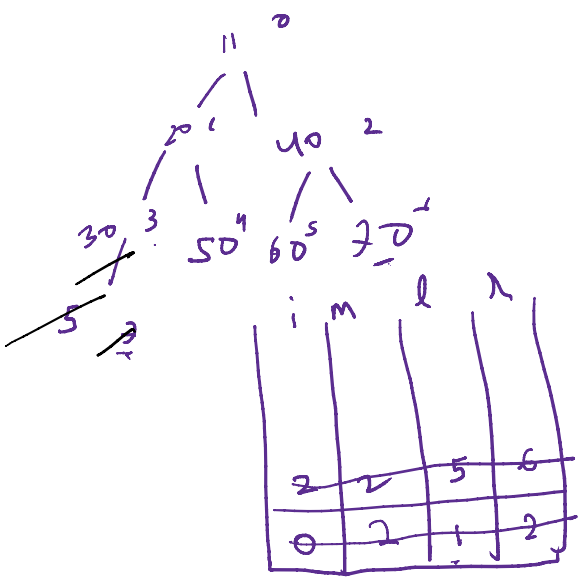





```
private void upHeapify(int i) {
    int pi = (i - 1) / 2;
    if (list.get(pi) > list.get(i)) {
        swap(i, pi);
        upHeapify(pi);
    }
}
```

i: 7

Downheapify




```
private void downHeapify(int i) {
    int minIndex = i;
    int li = 2 * i + 1;
    if (li < list.size() && list.get(li) < list.get(minIndex)) {
        minIndex = li;
    }
    int ri = 2 * i + 2;
    if (ri < list.size() && list.get(ri) < list.get(minIndex)) {
        minIndex = ri;
    }
    if (i != minIndex) {
        swap(i, minIndex);
        downHeapify(minIndex);
    }
}
```

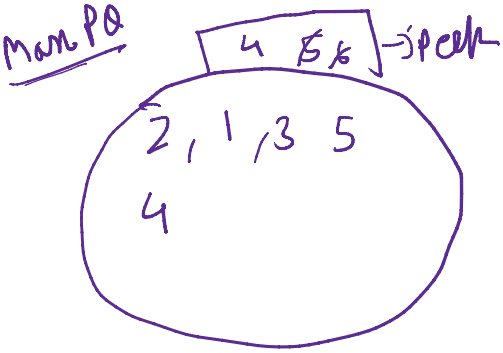
Kth Largest Element:.

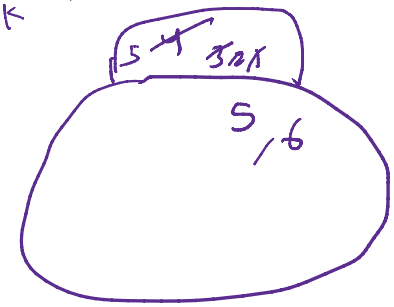
[3, 2, 1, 5, 6, 4] =)

K = 2

Size = 6

K = largest = 5



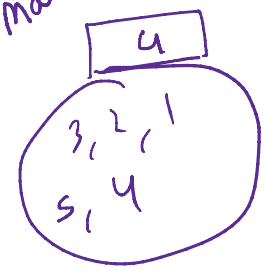


1, 2, 3, 4

[... 5, 4]

[3, 2, 1, 5, 6, 4]

maxPQ



k = 2 X 0

res = ~~6~~ 5

```java
public int findKthLargest(int[] nums, int k) {
    PriorityQueue<Integer> pq=new PriorityQueue<>(Collections.reverseOrder());
    for(int i=0;i<nums.length;i++){
        pq.add(nums[i]);
    }
    int res=0;
    while(k>0){
        res=pq.remove();
        k--;
    }
    return res;
}
```

H.W =) https://practice.geeksforgeeks.org/problems/merge-k-sorted-arrays/1

9-11

K = 3

0 [[1, 2, 3],
1 [2, 4, 6],
2 [5, 6, 7]]

PQ

1, 2, 2, 3, 4, 5, 6, 6, 7