

Interview

prime

→ codeX

Questions (102

Questi

102 ✓

Interview

MCT

4

1 1 0 0

0 1 0 1

→ Student (0, 1)

↑

S+

one's = 2

zero's = 2

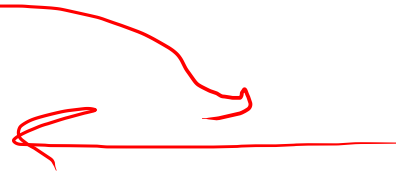
0

0

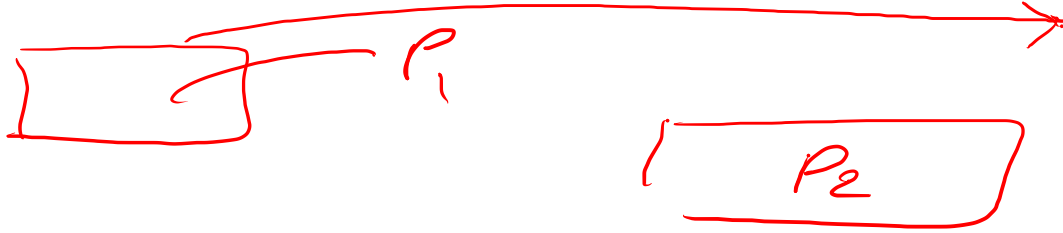
0

# Priority Queue

Emergency



Serious



Queue  $\rightarrow$  Interface

Priority queue  $\rightarrow$  class

Heap data structure

1

3

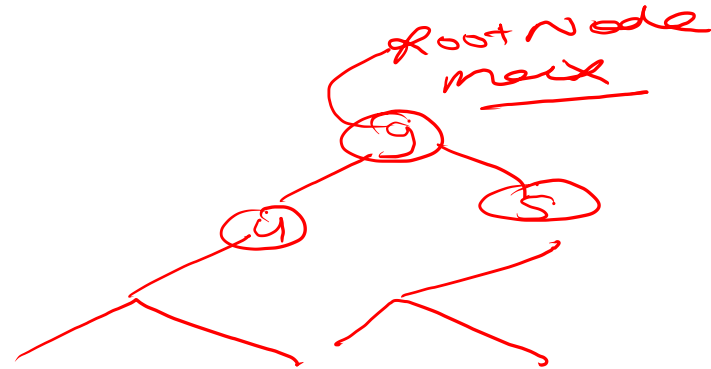
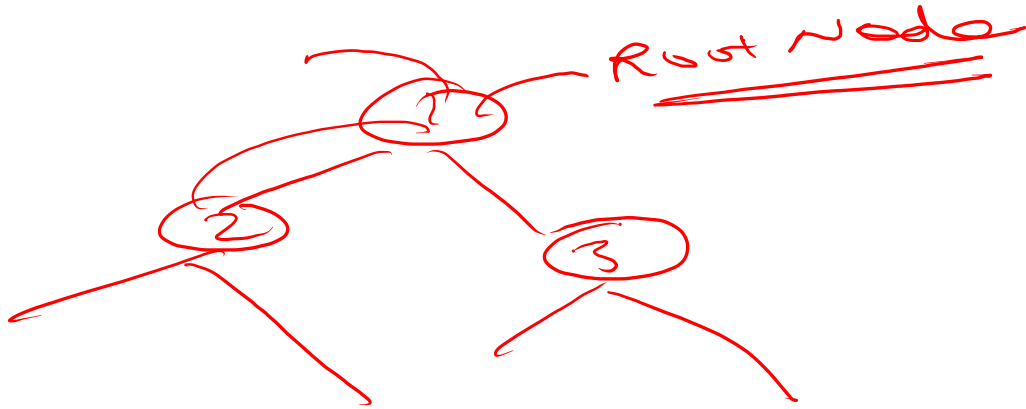
5

7

5x50 (FPS)

① MIN Heap

② max heap



Queue

+ Linked list()

+ Array Deque

+ Priority queue

find out  $k^{\text{th}}$  largest value

$\{20, 10, 17, 18, 15\}$

$17$  →

$k-1$

$k=3$

$k=$

22

⑦  $\{20, 18, 17, 15, 10\}$

It is important to note that the elements of a priority queue may not be sorted. However, elements are always retrieved in sorted order.

6

2 7 4 1 8 1

~~+~~ ~~A~~ ~~X~~

$$\begin{array}{cc} x & y \\ 7 & 8 \end{array}$$

$$\boxed{8-7} = 1$$

$$[2, 7, 4, 1, 8, 1]$$

$$x=2 \quad y=4$$

$$[2, 4, 1, 1, 1]$$

$$4-2=2$$

$$[1, 1, 1, 2]$$

$$2-1=1$$

$$[1, 1, 1]$$

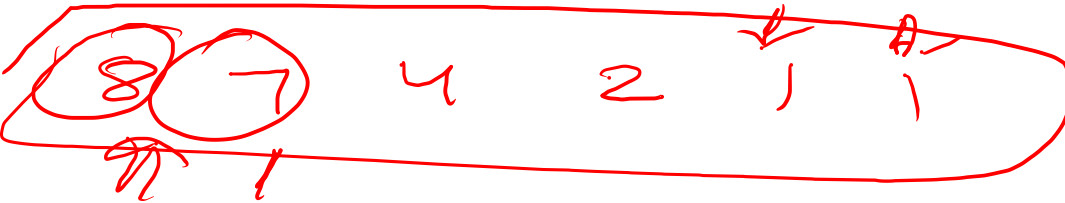
~~X~~ ~~X~~

$$[1]$$



6  
2 7 4 1 8 1 → 902

Pr. rem



8 7

if ( $y > x$ )

Pr. add ( $y - x$ )

# Break stone

```
6
7 public class Solution {
8
9     public static void main(String[] args) {
10         /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
11         Scanner sc = new Scanner(System.in);
12         int n = sc.nextInt();
13         ArrayList<Integer>arr = new ArrayList<>();
14         for(int i=0;i<n;i++){
15             arr.add(sc.nextInt());
16         }
17         // System.out.println(arr);
18         int ans = breakStone(arr);
19         System.out.println(ans);
20     }
21     public static int breakStone(ArrayList<Integer> arr){
22         // System.out.println(arr);
23         // while(arr.size()>1){
24         //     Collections.sort(arr);
25         //     int size = arr.size();
26         //     int temp = arr.get(size-1) - arr.get(size-2);
27         //     arr.remove(size-1);
28         //     arr.remove(size-2);
29         //     if(temp!=0){
30         //         arr.add(temp);
31         //     }
32         // }
33
34         PriorityQueue<Integer> pq = new PriorityQueue<>(Collections.reverseOrder());
35         for(int x: arr){
36             pq.add(x);
37         }
38         int x=0;
39         int y=0;
40
41         while(pq.size()>1){
42             y = pq.remove();
43             x = pq.remove();
44
45             if(y>x){
46                 pq.add(y-x);
47             }
48         }
49
50         return pq.size()==1?pq.poll():0;
51     }
```

Increasing

Decreasing