PW SKILLS

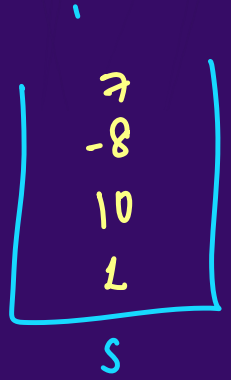
Priority Queues

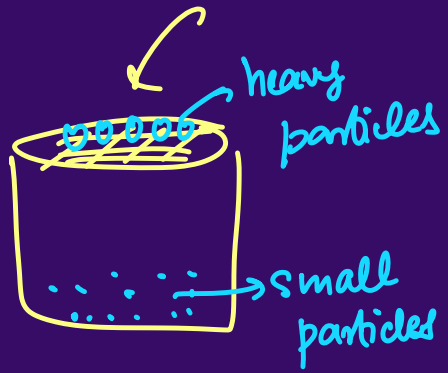
What and Why?

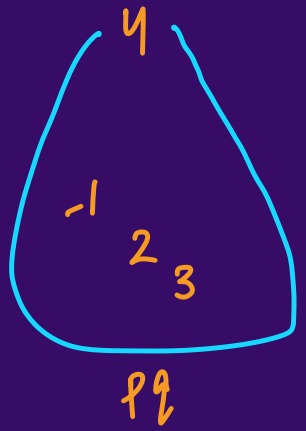
BST → searching → O(logn) T.C

insertion →

push, pop, top








```
pq.pop();
cout<<pq.top();
```

10

What and Why?

T.C.

top() → O(1)

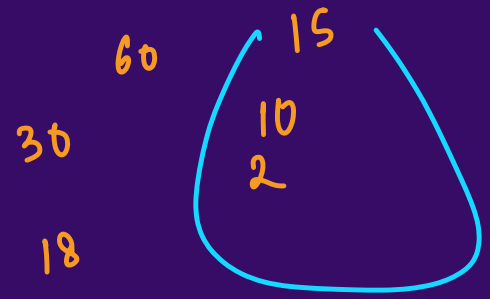
push(x) → O(log n)

pop() → O(log n)

current size ↑ 1

If we want a DS in which we can always get the max^m or min^m element at any pt. of time then we use pqueues (heaps)

n = pq.size();



push(10)	pop()
push(2)	top() → 15
push(18)	push(60)
pop()	pop
push(30)	
push(15)	

2 Types of heaps

max Heap (default)

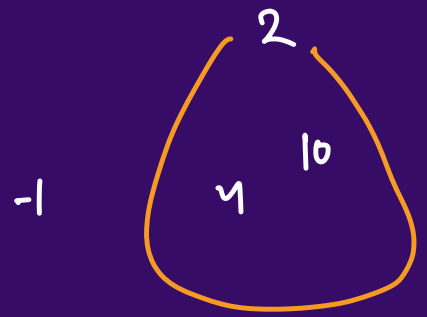
↓

max ele is on the top

min Heap

↓

min element is on top



push(10)
push(2)
push(4)
push(-1)
pop()

Priority Queue STL

PW SKILLS

```
#include <queue>
```

maxHeap → `priority_queue<int> pq`

minHeap → `priority_queue<int, vector<int>, greater<int>> pq`

Problem Identification

1) K^{th} smallest, Largest . Top K frequent elements, Closest K

2) At any pt. of time, minimum/max elements are required

3) Sorting (sometimes)

Ques:

PW SKILLS

Q1 : Find the kth smallest element in a given array.

arr = { 10, 20, -4, 6, 18, 24, 105, 118 } k = 3

Method-1 : BI sort , arr[k-1] T.C. = O(n log n) S.C. = O(log n)

Method-2 : Selection Sort → T.C. = O(k*n) S.C. = O(1)

Method-3 : Quick Select → T.C. = O(n) [Not in the worst case]

Method-4 : Using Heap → T.C. = O(n · log K)

Ques:

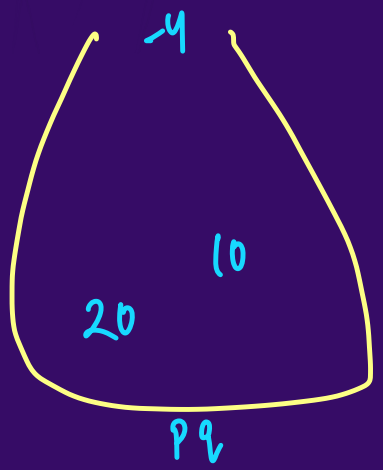
Q1 : Find the kth smallest element in a given array.

minHeap

arr = { 10, 20, -4, 6, 18, 24, 105, 118 } K = 3

```
for(int i=1; i<K; i++) pq.pop();
cout << pq.top()
```

→ K · log n



24 118 105 18 6

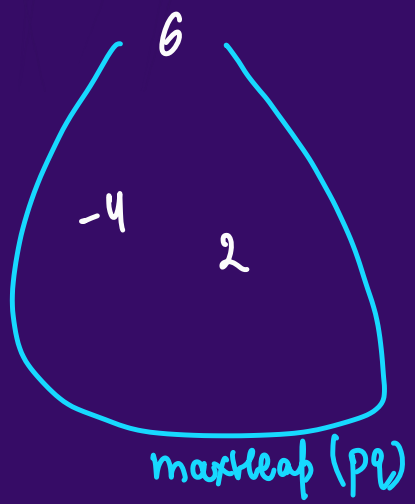
Ques:

maxHeap → only k elements not more than that

PW SKILLS

Q1 : Find the kth smallest element in a given array.

arr = { 10, 20, -4, 6, 18, 2, 105, 118 } K = 3



20 18 10 105 118

Ques:

PW SKILLS

Q1 : Find the kth smallest element in a given array.

```
for(int i=0;i<n;i++){
    pq.push(arr[i]); → log k
    if(pq.size()>k) pq.pop(); → log k
}
```

Time & Space

T.C. = $O(n * \log k)$

S.C. = $O(n)$ (total space)

✓ A.S. = $O(k)$

If I insert 'n' elements in a heap,

$$\log(1) + \log(2) + \log(3) \ldots \log(n)$$

$$\Rightarrow \log(n!) \rightsquigarrow n \log n$$

Ques:

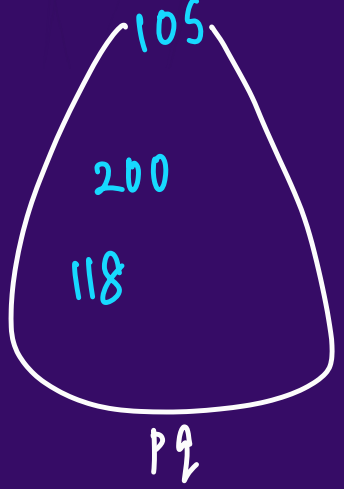
minHeap ↑

Q2 : Find the kth largest element in a given array.

arr = { 10, 200, -4, 6, 18, 2, 105, 118 } K = 3

T.C. = O(n log k)

S.C. = O(K) (Extra)



-4 6 2 10 18

[Leetcode 215]

Ques:

PW SKILLS

Q3 : Sort a 'k' sorted array (sort a nearly sorted array).

{ 4, 7, 8, 9, 10, 50, 60, 70 }
↑ sort

arr = { 10, 9, 8, 7, 4, 70, 60, 50 } k = 4

arr = { 6, 5, 3, 2, 8, 10, 9 } k = 3
↓ sort
{ 2, 3, 5, 6, 8, 9, 10 }

minHeap

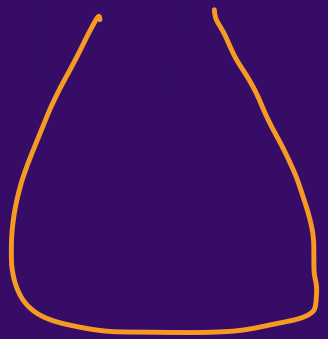
Ques:

PW SKILLS

Q3 : Sort a 'k' sorted array (sort a nearly sorted array).

arr = { 6, 5, 3, 2, 8, 10, 9 } k = 3

ans = { 2 3 5 6 8 9 10 3



PQ (minHeap)
size <= K

H.W. Leetcode 378
k^{th} smallest element in a sorted Matrix

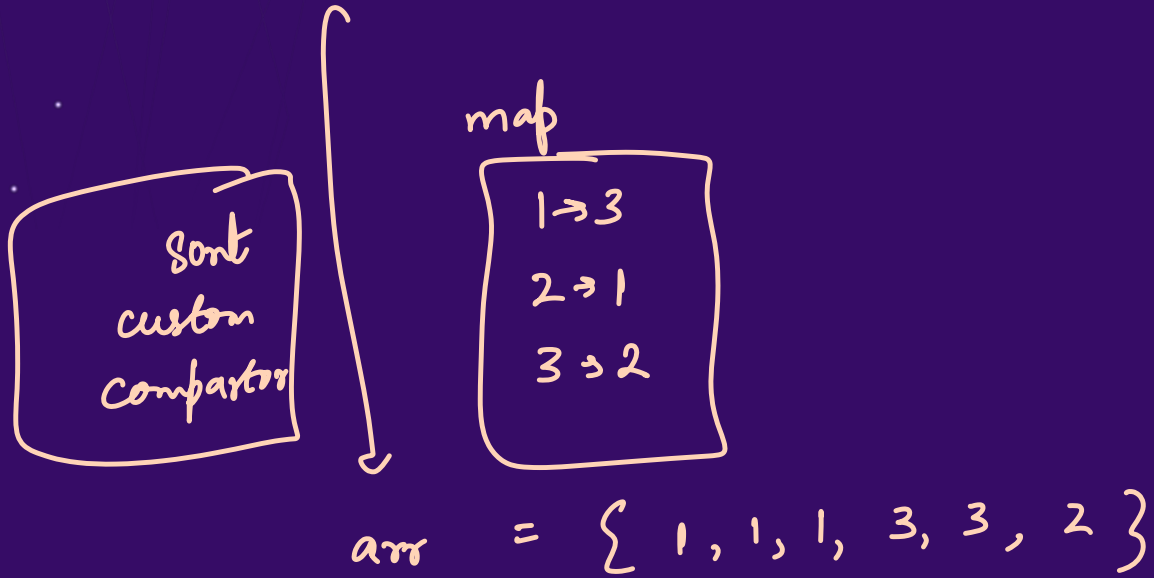
Ques: pair ka usage map

SKILLS

Q4 : Top K Frequent Elements

arr = {1, 3, 2, 1, 1, 3} K=2

ans = {1, 3}



[Leetcode 347]

Ques: pair ka usage map

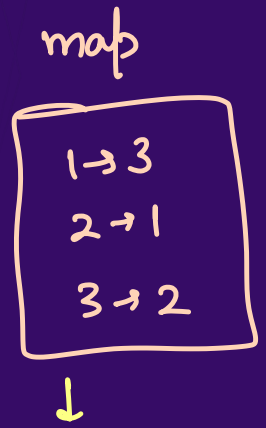
PW SKILLS

Q4 : Top K Frequent Elements

K largest elements

largest [as in with most frequencies] → minheap

arr = {1, 3, 2, 1, 1, 3} K=2 ans = {1, 3}



pair < ele, freq >

→ pair < freq, ele >

(1, 2)

[Leetcode 347]

Homework:

Q : Sort Array by Increasing Frequency.

[Leetcode 1636]

Ques:

Q5 : Find K Closest Elements

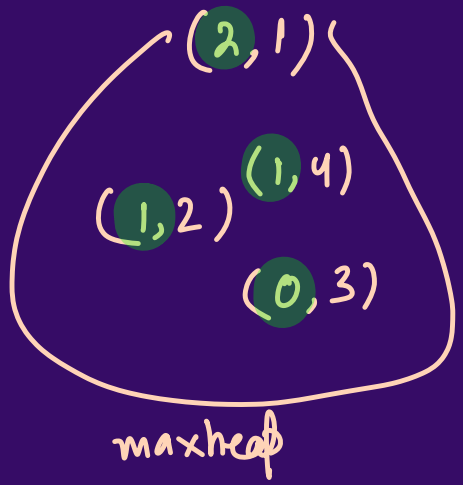
$x = 3$ arr = {1, 2, 3, 4, 5, 6} $K = 4$

2 1 0 1 2 3

K smallest distances

(3, 6)

(2, 5) VVIMP



heap → < dist, ele >

[Leetcode 658]

Ques:

Q6 : K Closest Points to Origin

arr = {{3,3}, {5,-1}, {-2,4}} K=2

$\sqrt{18}$ $\sqrt{26}$ $\sqrt{20}$

K smallest distances

maxheap

ans = {{3,3}, {-2,4}}

heap → <dist², point>

pair < int , pair<int,int> >

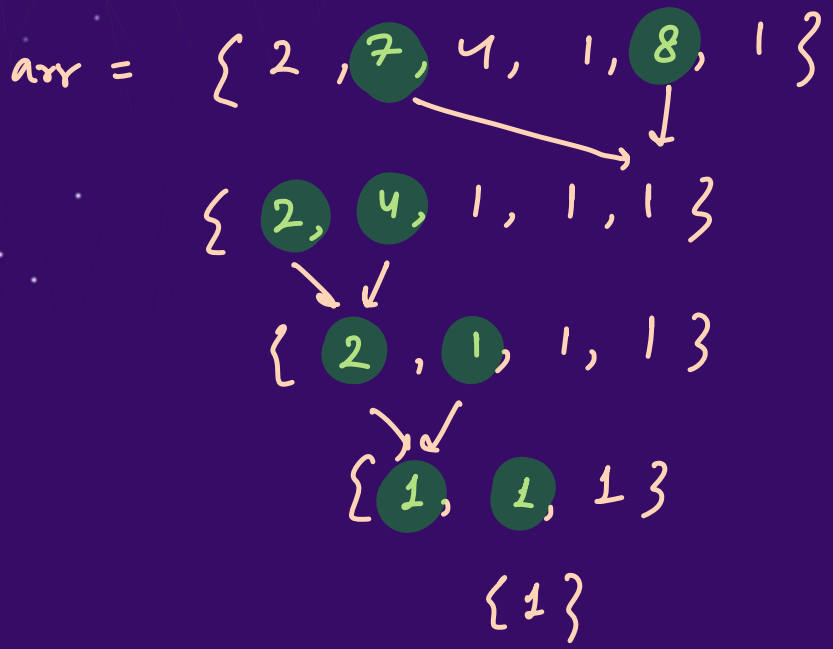
vector<int>

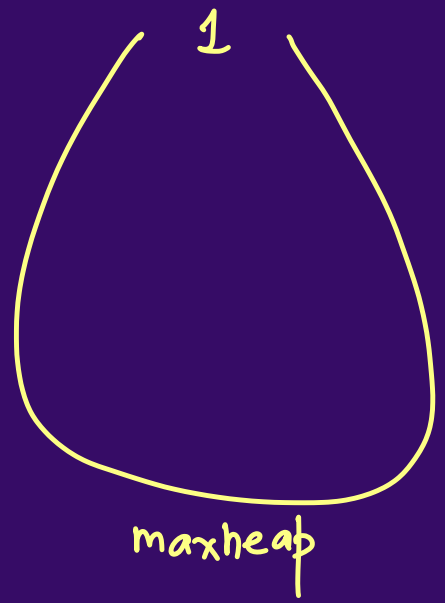
{18, {3,3}}

[Leetcode 973]

Ques:

Q7 : Last Stone Weight





x = 1

y = 1

x-y = 0

[Leetcode 1046]

Ques:

Q7 : Last Stone Weight

arr = {2, 7, 4, 1, 8, 1}

sort

{1, 1, 2, 4, 7, 8}

{1, 1, 2, 4, 1}

sort

{1, 1, 1, 2, 4}

{1, 1, 1, 2}

sort

{1, 1, 1, 2}

{1, 1, 1} → sort

{1}

↑

{1, 1, 1}

$n \log n + (n-1)\log(n-1) + (n-2)\log(n-2)$

$$\sum_{r=0}^{n-1} (n-r)\log(n-r)$$

[Leetcode 1046]

Ques: 'Very Interesting'

PW SKILLS

Q8 : Minimum Cost to Connect all Ropes

arr = { 2, 7, 4, 1, 8 }

9, 4, 1, 8

13, 1, 8

14, 8

22

cost = 9 + 13 + 14 + 22 = 58

task is to join all the ropes with the min cost possible

You can connect only 2 ropes at a time with the cost being sum of length of those ropes

Ques:

Q8 : Minimum Cost to Connect all Ropes

arr = { 2, 7, 4, 1, 8 }

total → $^{n}C_2 \times {}^{n-1}C_2 \times \cdots {}^{2}C_2$

6, 7, 1, 8

6, 8, 8

6, 16

22

cost = 6 + 8 + 16 + 22 = 52

Ques:

Q8 : Minimum Cost to Connect all Ropes

arr = {2, 7, 4, 1, 8}

{3, 7, 4, 8}

{7, 7, 8}

{14, 8}

cost = 3 + 7 + 14 + 22 = 46

Ques:

Q8 : Minimum Cost to Connect all Ropes

arr = { 6, 5, 3, 2, 8, 10, 9 }

cost = 5 + 10 + 14 + 19 + 24 + 43

THANK YOU