295 Find Median for Data streams

Page No.
Date :

1) P ⇒ [1,2,3]
0) P ⇒ [1.5,2]

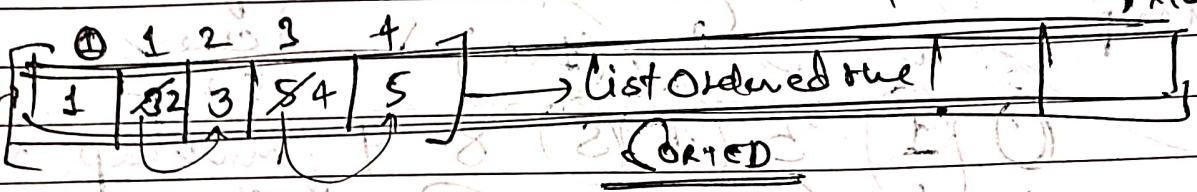
Median = Ordered list → middle element

Odd ∈ | 2 (5) 7 8

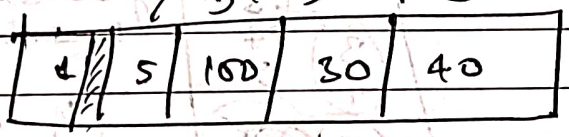
Even | (2+7) 8
= 3.5

Anymoment you have to give a median.

A1) Brute force — Given: Almost 5×10⁴ calls will be made to addNum & find Median.



1, 3, 2, 5



TLE — Large no. of elements

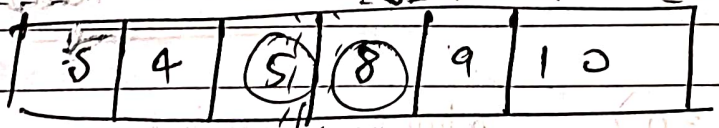
2

A2) PriorityQueue (Heap) ⇒ Don't want to sort the list everytime when short element comes!

nlogn

That's why we need maxheap so that it give sorted way.

n=6

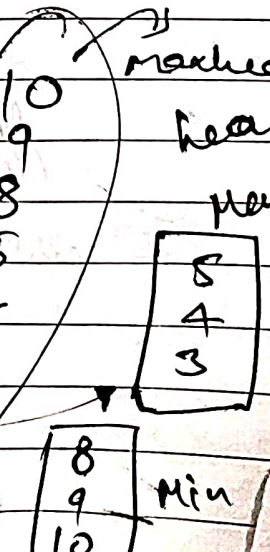


even case

PQ1 - maxheap — PQ2 - minheap

Why we are doing this?

Because we need that (5) in topmost! O(1)



O(1) .top() + .top()

maxheap | minheap

5
4
3

8
9
10

2

n = odd

3	4	5	8	9

Assuming that

5
4
3

maxheap

8
9

minheap

.top()

After inserting 4

(1<4)

check sizeof minheap() should not greater than maxheap()

(4-2>1)

0 1 3 4 5 8 9

(0<3) (1<5)

3
1
0

maxheap

0<1

left (i++)

.top()

maxheap

3
4
5

3
4

maxheap

At this moment minheap() no.of element > maxheap()

minheap().pop().push(maxheap)

find median

(5)/2 != 0) return maxheap.top()

else

int a = maxheap()

b = minheap()

(a+b)/2;

if (minheap > maxheap)

PQ <> L;
PQ <> R;

Page No.
Date :

MedianFinder(){
//maxheap L = new PQ<>(Collection.reverseOrder());
//minheap R = ——— ""();
}

addNum(){
if (L.isEmpty() || ~~R.isEmpty()~~ num < L.peek()) {
L.offer(num);
else{ R.offer(num);

Leftheap
//maxheap
ke andar kab
push karna hai

minheap
and num > L.peek()

At this time R may have more elements L

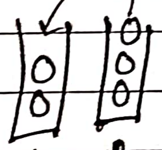
// Always maintain L.size() >= R.size()

(L ≯ R+1)

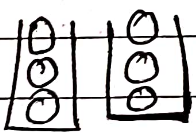
if (abs(L.size()) - abs(R.size())) > 1
R.push(L.top()); L.pop();
}elseif(L < R){ L.push(R)
size size R.pop()
}

L-R > 1
L > R+1

★★★★★
that means
L.size() must
not greater
than R.size()
+1


L R

double findMedian(){
}



even no. of element
if(L.size() == R.size()){
return (double)(~~R~~ R.top() + L.top()) / 2)
//odd case return L.top();

NOTE: Do remember to use # Offer, Peek,
Poll

Optimistic PQ
Remember Initialisation of PriorityQueue.