

Heaps Lecture 3

Today's checklist



7. Questions on heaps

Minneap /
$$\rightarrow$$
 pop() \rightarrow 0(log n)

Maxheap puru() \rightarrow 0(log n)

 $top() \rightarrow$ 0(1)

 $find(v) \rightarrow$ 0(n)

remove(x) \rightarrow 0(n) + 0(log n)

neglect



Q1: Find Median from Data Stream.



6, 1, 2, 4, 2, 1, 3, 8, 1



Q1: Find Median from Data Stream.

v = 1 2 6 4

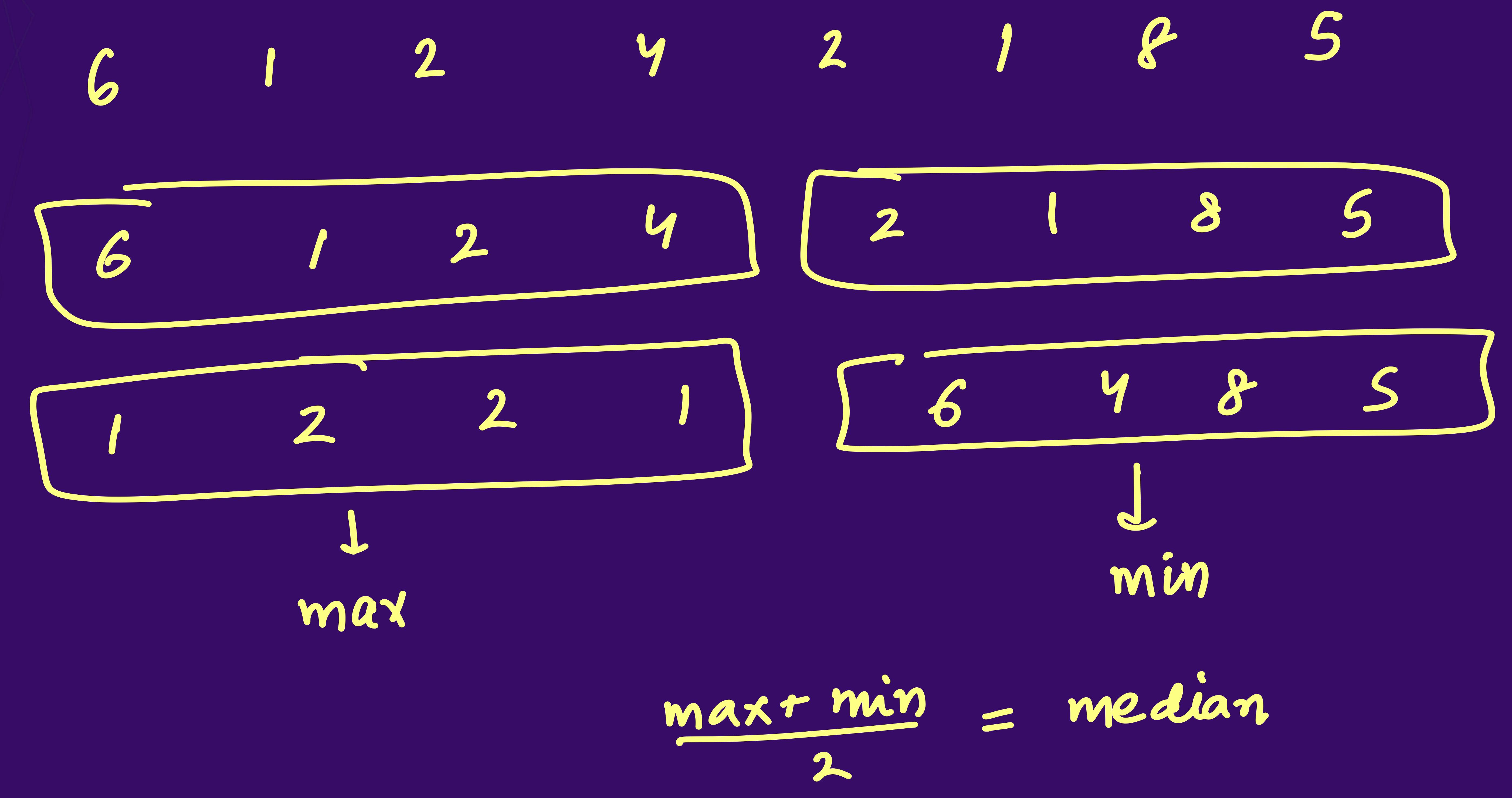
void addNum(int num) { Y. push_back(num)		
} Sort		
<pre>double findMedian() {</pre>		
}		

Stream	median
6	6
6	3-5
6 1 2	2
6 1 2 4	3

2 heaps, 1 maxheop, 1 minheap



Q1: Find Median from Data Stream.

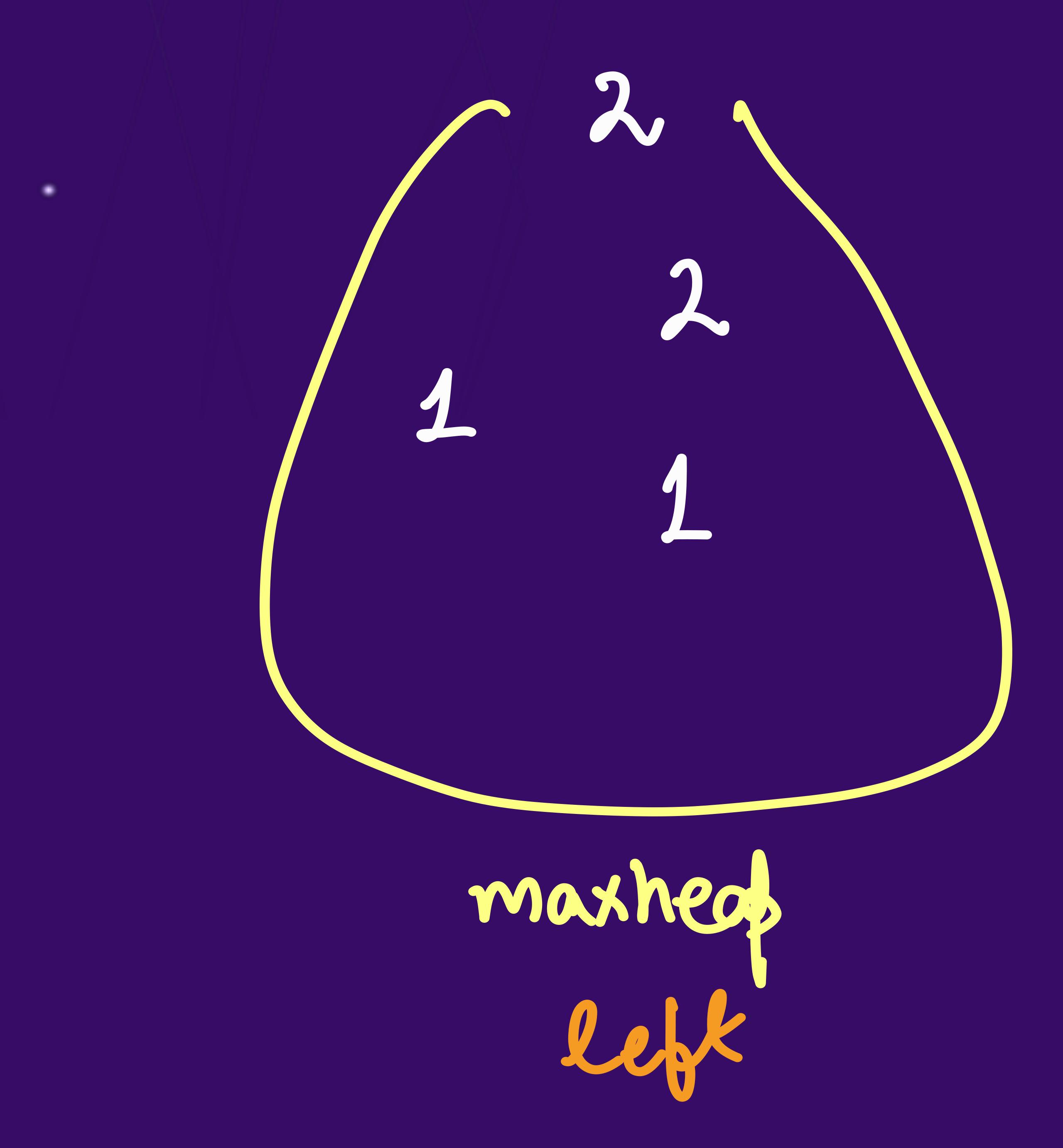


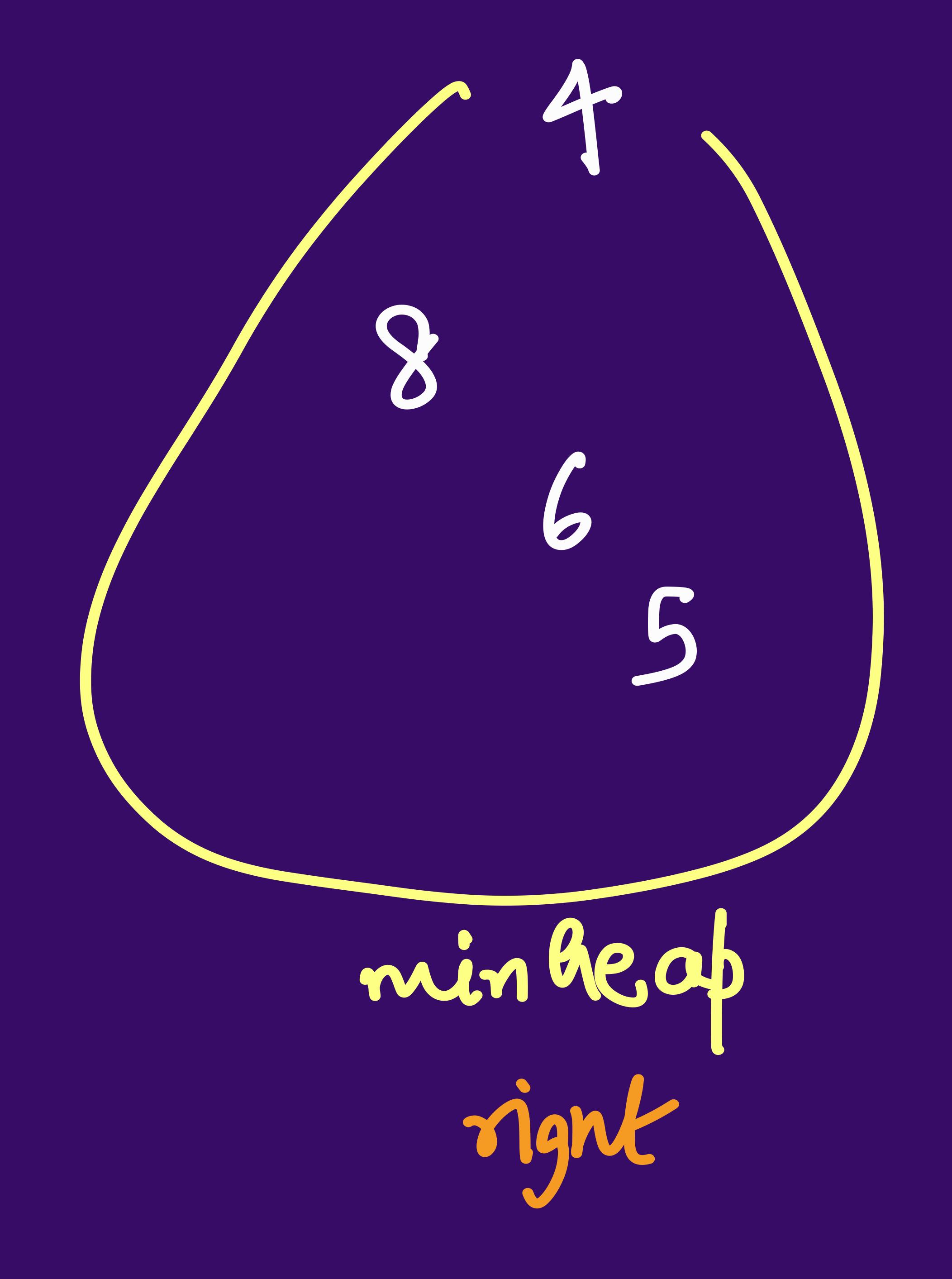
Oues:



Q1: Find Median from Data Stream.

6 1 2 4 5 6 3.5 2 3 2 2 3





- 1) all elements of left

 2 = all elements of
- 2) Size of lost b night should be same or tre diff Should be 1

```
priority_queue<int> left; // maxHeap
priority_queue<int, vector<int>, greater<int>> right; // minHeap
void addNum(int num) {b(loan)
   if(left.size()==0 || num<left.top()) left.push(num); Lean.</pre>
    else right.push(num);
    if(left.size()>right.size()+1){
       right.push(left.top()); Log-
        left.pop();
    if(right.size()>left.size()+1){
       right.pop(); (10)
double findMedian() { // 0(1)
    if(left.size()==right.size())
       return (left.top() + right.top())/2.0;
    else
       if(left.size()>right.size()) return left.top();
       else return right.top();
```





Q2: Smallest Range covering elements from K Lists

$$\begin{cases} 9, 10, 15, 24, 263 \\ 50, 9, 12, 203 \\ 5, 18, 22, 303 \end{cases}$$

$$[0,30]$$
 30 $[10,18]$ 8 $[0,5]$ 5 $[4,4]$ 5

[Leetcode 632]

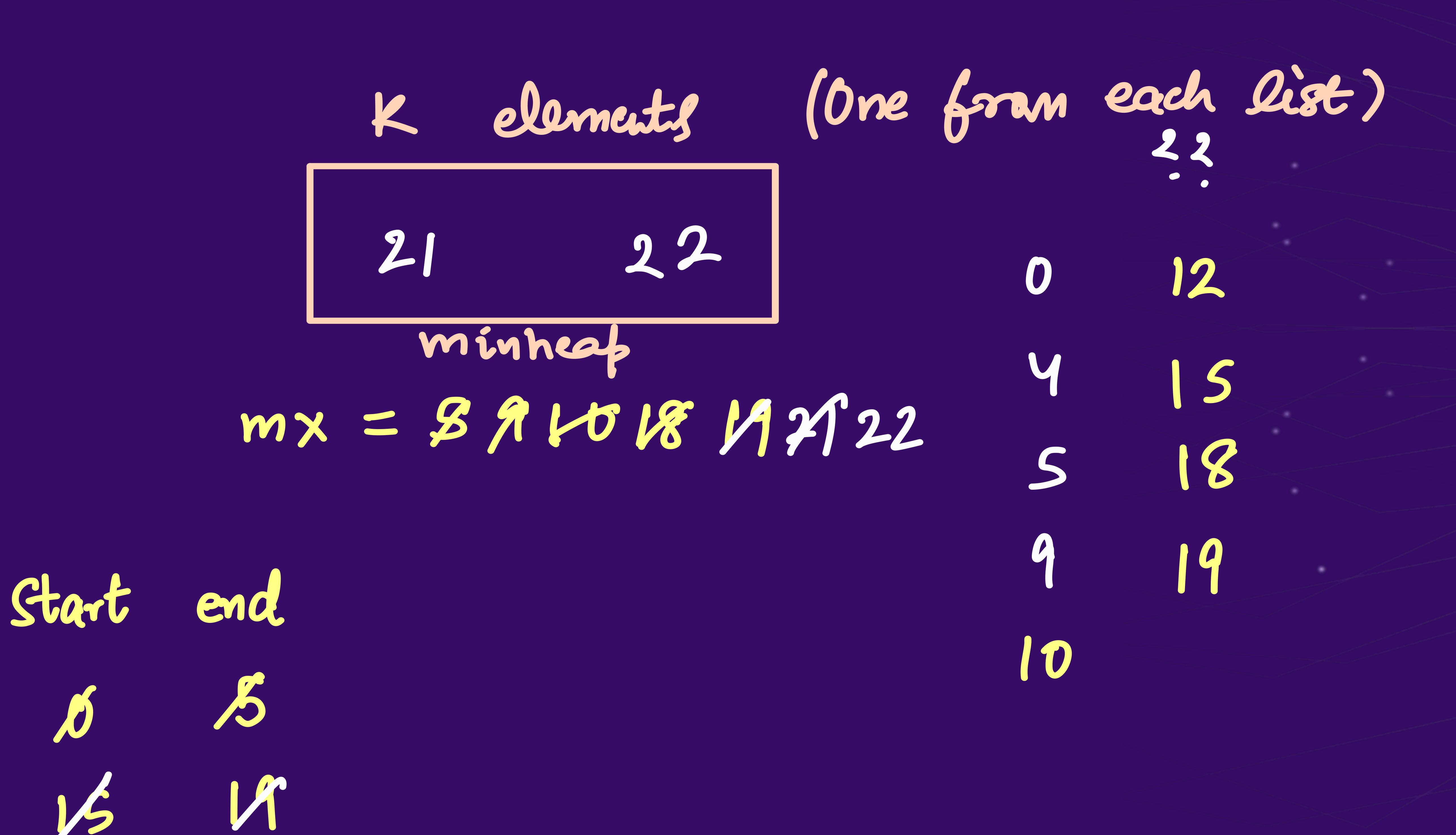
Heap element



num, row, col

Q2: Smallest Range covering elements from K Lists

$$\begin{cases} 4, 10, 15, 21, 263 \\ 5, 9, 12, 193 \end{cases}$$



pair < int, pair < int, mt??

18 21

[Leetcode 632]

```
typedef pair<int,pair<int,int>> pip;
vector<int> smallestRange(vector<vector<int>>& arr) {
   priority_queue<pip,vector<pip>,greater<pip>>> pq;
   // pq element -> {arr[row][col],{row,col}}
   int mx = INT_MIN;
   for(int i=0;i<arr.size();i++){</pre>
       mx = max(mx, arr[i][0]);
       pq.push({arr[i][0],{i,0}}); -> 0 K LogK
   int mn = pq.top().first;
   int start = mn, end = mx;
   while(true){
       int row = pq.top().second.first;
       int col = pq.top().second.second;
       pq.pop();
       if(col==arr[row].size()-1) break;
       pq.push({arr[row][col+1],{row,col+1}});
                                                      o(nk legk)
       mx = max(mx, arr[row][col+1]);
       mn = pq.top().first;
       if(mx-mn < end-start){</pre>
           end = mx;
           start = mn;
   return {start,end};
```

K lists each of size 'n' on an average ng elements from K Lists

```
T.C. = O(nklogk)

= O(mlogk)

L

m is the total

no. of elements
```

[Leetcode 632]

B SKILLS

pair (int, pair zint, int >> P = { 0, {1, 43}}

p. first p. second. first

THANKYOU