

$C \rightarrow L$

$a \rightarrow L$

$b \rightarrow L$

$hfc = a$

2<sup>nd</sup>

2<sup>var</sup>

pay

this - 0  
0 - this

compare

Heap  $\Rightarrow$  Priority Queue

Queue  $\times$   
FIFO  $\times$

operations

add() } O(log n)  
remove() }

peek() }  $\Rightarrow$  O(1)  
size() }

processing  $\rightarrow$  priority

priority  $\rightarrow$  (min priority) (rank list)

(max priority) (total score)

By Default = min - priority

Integer

PriorityQueue<Integer> pq = new PriorityQueue<>(Collections.reverseOrder());

Inverts  
Notime  $\Rightarrow$  Collections.reverseOrder()

PriorityQueue <> Pq = new PriorityQueue();

Example

PriorityQueue<Integer> pq = new PriorityQueue();

pq.add(10);

pq.add(5);

pq.add(20);

pq.add(-1);

// -1 5 10 20

// 2407 1998

$\rightarrow$

2407 1998 % 10000  $\Rightarrow$

1998

```


PriorityQueue<Integer> pq = new PriorityQueue<>();

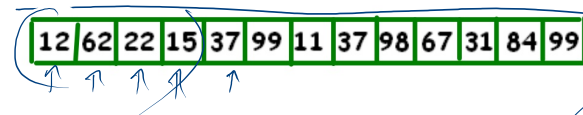
for(int i = 0 ; i < k ; i++){
    pq.add(arr[i]);
}

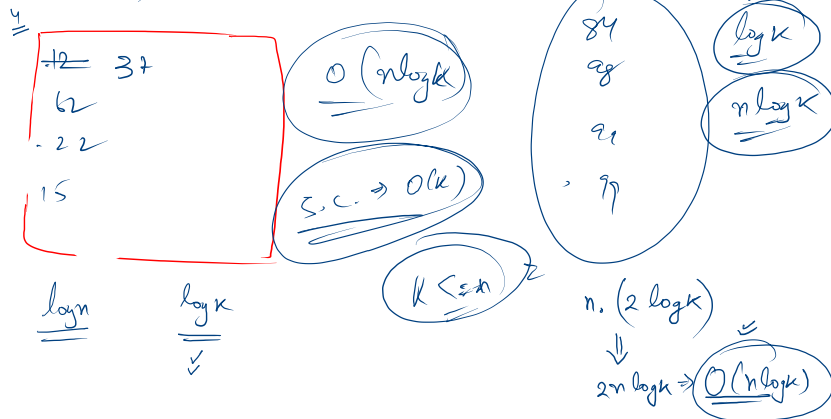
for(int i = k ; i < n ; i++){
    if(arr[i] > pq.peek()){
        pq.remove();
        pq.add(arr[i]);
    }
}

while(pq.size() > 0){
    System.out.println(pq.remove());
}

```

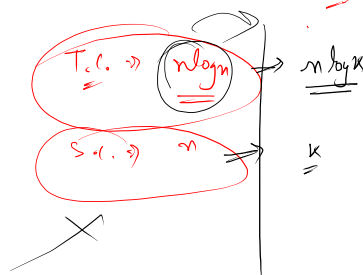
K=4  
Min Priority  


12 62 22 15 37 99 11 37 98 67 31 84 99  


log n log k  


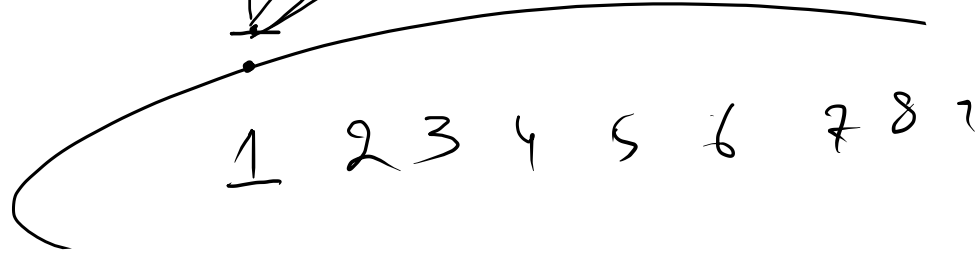
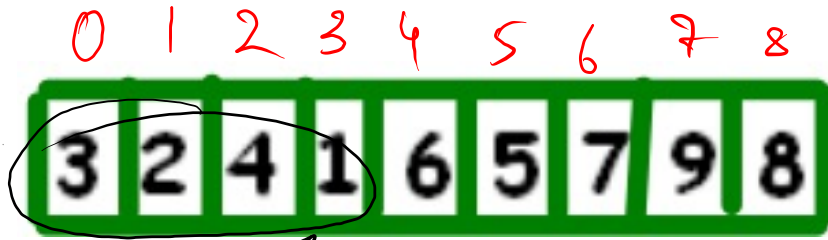
Max Priority  
n

12 99 31  
62 11 ~~84~~  
22 37 ~~27~~  
15 ~~98~~ ~~99~~  
37 67

1 2 3 4  
↓ ↓ ↓ ↓  
99 99 98 87  
  
T.C. -> n log n  
S.C. -> n  
k

ele  $\Rightarrow$  atmost  $K$  idx shifted  
from sorted  
pos it's

$K=3$



3	2	4	1
6	5	7	9
8			

$k=3$

~~3~~ ~~2~~ ~~4~~  
~~1~~ ~~6~~  
~~5~~ ~~7~~  
~~8~~ ~~9~~

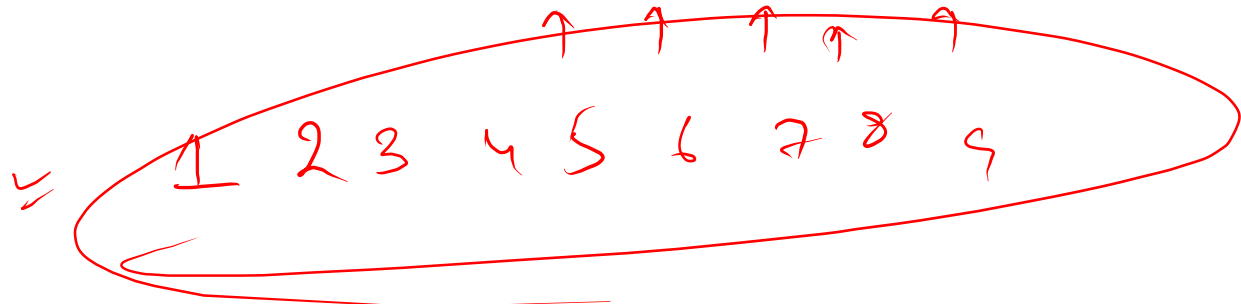
```
✓ PriorityQueue<Integer> pq = new PriorityQueue<>();
```

```
for(int i = 0 ; i <= k ; i++){  
    pq.add(arr[i]);  
}
```

```
✓ for(int i = k+1 ; i < n ; i++){  
    System.out.println(pq.remove());  
    pq.add(arr[i]);  
}
```

```
{ while(pq.size() > 0){  
    System.out.println(pq.remove());  
}
```

0	1	2	3	4	5	6	7	8
3	2	4	1	6	5	7	9	8

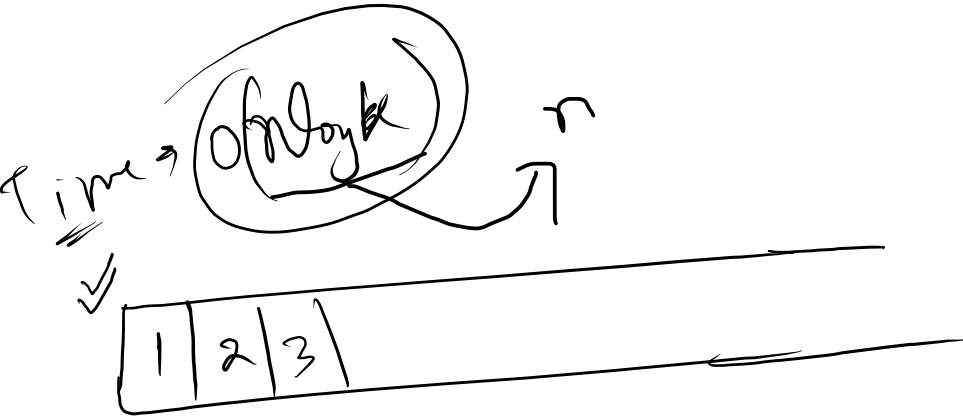


O(N)

lists =

→

0	10	20	30	40	50		
1	5	7	9	11	19	55	57
2	1	2	3				
3	32	39					



tmp = 2-2-3

0-0-10

1-0-5

3-0-32

Pair < Li, vi, val >

```
ArrayList<Integer> rv = new ArrayList<>();
PriorityQueue<Pair> pq = new PriorityQueue<>();

for(int i = 0 ; i < lists.size() ; i++){
    pq.add(new Pair(i,0, lists.get(i).get(0)));
}

while(pq.size() > 0){
    Pair tmp = pq.remove();

    rv.add(tmp.val);

    if(tmp.vi+1 < lists.get(tmp.li).size() ){
        tmp.vi = tmp.vi+1;
        tmp.val = lists.get(tmp.li).get(tmp.vi);
        pq.add(tmp);
    }
}

return rv;
```