**Lab 6.**

Solve these problems for the priority queue.

Deadline: Week 8.

<https://leetcode.com/problems/kth-largest-element-in-an-array/>

class Solution {

public int findKthLargest(int[] nums, int k) {

Arrays.sort(nums);

return nums[nums.length-k];

}

}

<https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/>

class Solution {

public int kthSmallest(int[][] matrix, int k) {

Queue<Integer> q = new PriorityQueue<>();

for (int i=0; i<matrix.length; i++) {

for (int j=0; j<matrix[i].length; j++) {

q.offer(matrix[i][j]);

}

}

while(--k>0) {

q.poll();

}

return q.poll();

}

}

<https://leetcode.com/problems/top-k-frequent-elements/>

class Solution {

static class Pair implements Comparable<Pair>{

int first;

int second;

Pair(int first, int second){

this.first= first;

this.second= second;

}

public int compareTo(Pair o){

return this.first-o.first;

}

}

public int[] topKFrequent(int[] arr, int k) {

HashMap<Integer, Integer> map = new HashMap<>();

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

int ans[]= new int[k];

for(int a: arr)

map.put(a, map.getOrDefault(a,0)+1);

for(Map.Entry<Integer,Integer> e: map.entrySet()){

pq.add(new Pair(e.getValue(), e.getKey()));

if(pq.size()>k)

pq.remove();

}

int i=0;

while(!pq.isEmpty()){

ans[i] = pq.remove().second;

i++;

}

return ans;

}

}

<https://leetcode.com/problems/top-k-frequent-words/>

class Solution {

public List<String> topKFrequent(String[] words, int k) {

HashMap<String, Integer> h=new HashMap<String,Integer>();

int count=0;

for(int i=0;i<words.length;i++)

{

if(!h.containsKey(words[i]))

{

h.put(words[i],1);

count++;

}

else

{

h.put(words[i],h.get(words[i])+1);

}

}

int arr[]=new int[count];

String str[]=new String[count];

int l=0;

for(Map.Entry<String,Integer> m:h.entrySet())

{

arr[l]=m.getValue();

str[l]=m.getKey();

l++;

}

for(int i=0;i<count;i++)

{

for(int j=i+1;j<count;j++)

{

if(arr[i]<arr[j])

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

String t=str[i];

str[i]=str[j];

str[j]=t;

}

else if(arr[i]==arr[j])

{

if(str[i].compareTo(str[j])>0)

{

String t=str[i];

str[i]=str[j];

str[j]=t;

}

}

}

}

ArrayList<String> result=new ArrayList<String>();

for(int i=0;i<k;i++)

{

result.add(str[i]);

}

return result;

}

}