HW_Kth smallest element 21

Given an array arr[] and an integer K where K is smaller than size of array, the task is to find the Kth smallest element in the given array. It is given that all array elements are distinct.

Input Format

The first line integer N represent the size of array

The second line n integer represent element of an array

The third line integer K represent kth smallest element

Constraints

1 <= N <= 10^5

1 <= arr[i] <= 10^5

1 <= K <= N

Output Format

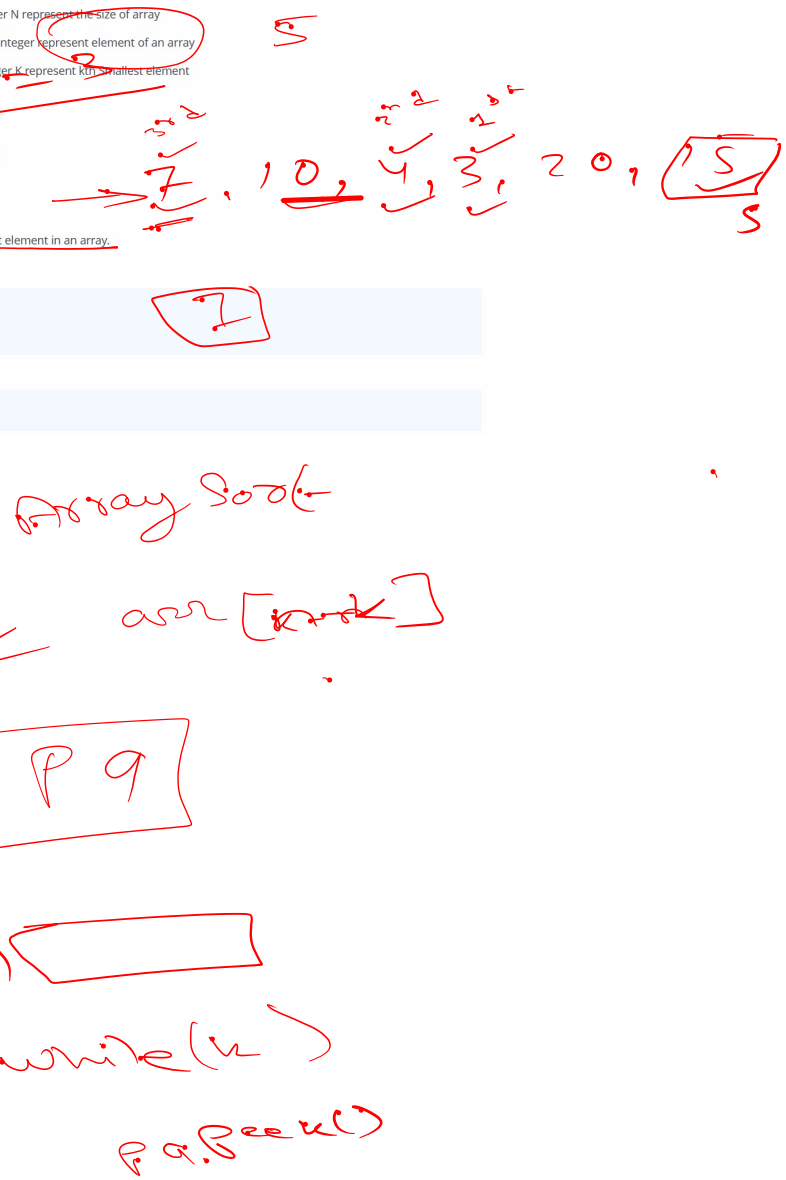
return kth smallest element in an array.

Sample Input 0

```
6
7 10 4 3 20 15
3
```

Sample Output 0

```
7
```



Submitted Code

Language: Java 8

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int [] arr = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();
        int k = sc.nextInt();

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

        for(int i=0;i<n;i++)pq.add(arr[i]);
        while(pq.size()>0 && k-1>0){
        pq.poll();k--;
        }
        System.out.print(pq.peek());
    }
}
```

HW_Relative Ranks

You are given an integer array score of size n, where score[i] is the score of the ith athlete in a competition. All the scores are guaranteed to be unique.

The athletes are placed based on their scores, where the 1st place athlete has the highest score, the 2nd place athlete has the 2nd highest score, and so on. The placement of each athlete determines their rank:

- The 1st place athlete's rank is "Gold Medal".
- The 2nd place athlete's rank is "Silver Medal".
- The 3rd place athlete's rank is "Bronze Medal".
- For the 4th place to the nth place athlete, their rank is their placement number (i.e., the xth place athlete's rank is "x").

Return an array answer of size n where answer[i] is the rank of the ith athlete.

Input Format

First line contains an input n.

Next line contains an integer array score[i] of size n.

Constraints

- n == score.length
- 1 <= n <= 104
- 0 <= score[i] <= 106
- All the values in score are unique.

Constraints

- n == score.length
- 1 <= n <= 104
- 0 <= score[i] <= 106
- All the values in score are unique.

Output Format

Print an array of strings answer of size n where answer[i] is the rank of the ith athlete.

Sample Input 0

```
5
5 4 3 2 1
```

Sample Output 0

```
Gold Medal Silver Medal Bronze Medal 4 5
```

Explanation 0

The placements are [1st, 2nd, 3rd, 4th, 5th].

Example 1:

Input: score = [5,4,3,2,1]
Output: ["Gold Medal","Silver Medal","Bronze Medal","4","5"]
Explanation: The placements are [1st, 2nd, 3rd, 4th, 5th].

Example 2:

Input: score = [10,3,8,9,4]
Output: ["Gold Medal","5","Bronze Medal","Silver Medal","4"]
Explanation: The placements are [1st, 5th, 3rd, 2nd, 4th].

10 , 3 , 8 , 9 , 4

ans [GM , 5 , BM , SM , 4]

10:37

Submitted Code

Language: Java 8

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int []arr = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();

        String ans [] = new String[n];

        PriorityQueue<Integer> pq = new PriorityQueue<>((a,b)->arr[b]-arr[a]);

        for(int i=0;i<n;i++)pq.add(i);

        int i=1;
        while(!pq.isEmpty()){
            int index = pq.poll();
            if(i==1)ans[index]= "Gold Medal";
            else if(i==2)ans[index]= "Silver Medal";
            else if(i==3)ans[index]= "Bronze Medal";
            else ans[index] =Integer.toString(i);
            i++;
        }

        for(int j=0;j<ans.length;j++){
            System.out.print(ans[j]+" ");
        }
    }
}
```

HW_maximum diamonds

There are N bags with diamonds in them. The i'th of these bags contains $A[i]$ diamonds. If you drop a bag with P diamonds, it changes to $[P/2]$ diamonds, and you gain P diamonds, where $[p]$ is the greatest integer less than p. Dropping a bag takes 1 minute. Find the maximum number of diamonds that you can take if you are given K minutes.

Input Format

- Two Integers N and K.
- N integers seperated by single space.

Constraints

```
1 <= N <= 10^5
0 <= K <= 10^5
0 <= A[i] <= 10^5
```

Output Format

Single Integer representing max diamonds.

Sample Input 0

```
5 3
2 1 7 4 2
```

Sample Output 0

```
14
```

Explanation 0

The state of bags is: **[2 1 7 4 2]**
You take all diamonds from Third bag (7).
The state of bags becomes: **[2 1 3 4 2]**
Take all diamonds from Fourth bag (4).
The state of bags becomes: **[2 1 3 2 2]**
Take all diamonds from Third bag (3).
The state of bags becomes: **[2 1 1 2 2]**
Hence, number of Diamonds = 7+4+3 = 14.

Submitted Code

Language: Java 8

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k= sc.nextInt();
        int [] arr = new int[n];
        for(int i=0;i<n;i++)arr[i]=sc.nextInt();


        PriorityQueue<Integer> pq = new PriorityQueue<>((a,b)->b-a);

        int sum =0;
        for(int i=0;i<n;i++){

            pq.add(arr[i]);
        }

        for(int i=0;i<k;i++){
            int max = pq.poll();
            sum+=max;
            int half = max/2;
           pq.add(half);
        }
        System.out.print(sum);
    }
}
```