**Maximum distinct elements after removing K elements**

Submissions: [1848](https://practice.geeksforgeeks.org/problem_submissions.php?pid=2867)  Accuracy:

37.06%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/0/0/)   Marks: 2

Show Topic Tags   

Given an array containing N elements. The task is to find maximum number of distinct elements after removing K elements from the array.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case consists of two lines. First line of each test case contains two Integers N and K and the second line contains N space separated elements.

**Output:**  
For each test case, print the maximum distinct elements after removing K elements in new line.

**Constraints:**  
1<=T<=100  
1<=K<=N<=106  
1<=A[i]<=105

**Example:  
Input:**  
2  
7 3  
5 7 5 5 1 2 2  
7 5  
1 2 3 4 5 6 7  
**Output:**  
4  
2

**Explanation:**

Input : A[] = {5, 7, 5, 5, 1, 2, 2}, K = 3

Output : 4

Remove 2 occurrences of element **5** and

1 occurrence of element **2**.

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/maximum-distinct-elements-after-removing-k-elements/0#ExpectOP) option \*\*

<https://practice.geeksforgeeks.org/problems/maximum-distinct-elements-after-removing-k-elements/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int MinimumRemove(int[] arr, int k)

{

Dictionary<int, int> frec =

new Dictionary<int, int>();

for (int i = 0; i < arr.Length; i++)

{

if (frec.ContainsKey(arr[i]))

{

frec[arr[i]]++;

}

else

{

frec[arr[i]] = 1;

}

}

var items = from pair in frec

orderby pair.Value descending

select pair;

Dictionary<int, int> nuevo =

new Dictionary<int, int>();

foreach (KeyValuePair<int, int> kvp in items)

{

nuevo[kvp.Key] = kvp.Value;

}

int cont = 0;

foreach (KeyValuePair<int, int> kvp in items)

{

if (cont < k)

{

if (kvp.Value > 1)

{

nuevo[kvp.Key] = 1;

cont += kvp.Value - 1;

}

//else

//{

// nuevo[kvp.Key] = kvp.Value;

//}

}

}

int[] keys = nuevo.Keys.ToArray();

for (int i = 0; i < keys.Length; i++)

{

if (cont < k)

{

if (nuevo[keys[i]] == 1)

{

nuevo[keys[i]] = 0;

cont++;

}

}

}

int ans = 0;

foreach (KeyValuePair<int, int> kvp in nuevo)

{

if (kvp.Value >= 1)

{

ans++;

}

}

return ans;

}

static void Main(string[] args)

{

//int[] arr = { 5, 7, 5, 5, 1, 2, 2 };

//int k = 3;

//Console.WriteLine(MinimumRemove(arr, k));

int t = int.Parse(Console.ReadLine().Trim());

while (t-- > 0)

{

string[] input = Console.ReadLine().Trim().Split(' ');

int n = int.Parse(input[0]);

int k = int.Parse(input[1]);

int[] arr = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

Console.WriteLine(MinimumRemove(arr, k));

}

//int[] arr = Array.ConvertAll("20 23 17 7 25 30 16 11 15 29 18 22 18 3 28 13 23 27 2 21 27 2 16 30 5 30 11 10 22 8 28 12 22 6 10 8 28 17 18 4 7 6 17 24 30 15 29 22 3 30 4 30 1 19 29 6 11 10 7 24 9 26 27 22 2 6 30 29 15 9 2 21 14 11 15 5 25".Split(' '), e => int.Parse(e));

//Console.WriteLine("\n" + MinimumRemove(arr, 64));

Console.ReadLine();

}

}

}