Non-Divisible Subset

by zxqfd555

Problem Submissions Leaderboard Discussions Editorial

Given a set, S, of n distinct integers, print the size of a maximal subset, S', of S where the sum of any 2 numbers in S' is *not* evenly divisible by k.

Input Format

The first line contains 2 space-separated integers, n and k, respectively.
The second line contains n space-separated integers (we'll refer to the i^{th} value as a_i) describing the unique values of the set.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq k \leq 100$
- $1 \leq a_i \leq 10^9$
- All of the given numbers are distinct.

Output Format

Print the size of the largest possible subset (S').

Sample Input

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

Sample Output

```
3
```

Explanation

The largest possible subset of integers is $S' = \{1, 7, 4\}$, because no two integers will have a sum that is evenly divisible by $k = 3$:

- $1 + 7 = 8$, and 8 is not evenly divisible by 3.
- $1 + 4 = 5$, and 5 is not evenly divisible by 3.
- $7 + 4 = 11$, and 11 is not evenly divisible by 3.

The number 2 cannot be included in our subset because it will produce an integer that is evenly divisible by $k = 3$ when summed with any of the other integers in our set:

- $1 + 2 = 3$, and $\frac{3}{3} = 1$ $(\textbf{remainder } 0)$.
- $4 + 2 = 6$, and $\frac{6}{3} = 2$ $(\textbf{remainder } 0)$.
- $7 + 2 = 9$, and $\frac{9}{3} = 3$ $(\textbf{remainder } 0)$.

Thus, we print the length of S' on a new line, which is **3**.

Submissions: 22098
Max Score: 20
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
class Solution
{

    static int maxNonDivisibleSubset(int[] s, int k)
    {
        int N = s.Length;
        int[] occ = new int[k];

        int res = 0;
        for (int i = 0; i < N; i++)
        {
            occ[s[i] % k]++;
        }//for
        for (int i = 1; i < (k + 1) / 2; i++)
            res += (occ[i] > occ[k - i] ? occ[i] : occ[k - i]);
        //res += (occ[0] >= 1) + (k % 2 == 0 && occ[k / 2]>0) );
        res += (occ[0] >= 1) ? 1 : 0;
        res += (k % 2 == 0 && occ[k / 2] > 0) ? 1 : 0;

        return res;

    }

    static void Main(String[] args)
    {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution
*/


        string[] nk = Console.ReadLine().Split(' ');
        int n = int.Parse(nk[0].ToString());
        int k = int.Parse(nk[1].ToString());

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

        Console.WriteLine(maxNonDivisibleSubset(a, k));


    }
}
```

Line: 42 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9
- ✔ Test Case #10
- ✔ Test Case #11
- ✔ Test Case #12
- ✔ Test Case #13
- ✔ Test Case #14
- ✔ Test Case #15

Next Challenge