Picking Numbers

locked

by wanbo

Problem Submissions Leaderboard Discussions Editorial

Given an array of integers, find and print the maximum number of integers you can select from the array such that the absolute difference between any two of the chosen integers is ≤ 1.

Input Format

The first line contains a single integer, n, denoting the size of the array.
The second line contains n space-separated integers describing the respective values of $a_0, a_1, \ldots, a_{n-1}$.

Constraints

- $2 \leq n \leq 100$
- $0 < a_i < 100$
- The answer will be ≥ 2.

Output Format

A single integer denoting the maximum number of integers you can choose from the array such that the absolute difference between any two of the chosen integers is ≤ 1.

Sample Input 0

```
6
4 6 5 3 3 1
```

Sample Output 0

```
3
```

Explanation 0

We choose the following multiset of integers from the array: $\{4, 3, 3\}$. Each pair in the multiset has an absolute difference ≤ 1 (i.e., $|4 - 3| = 1$ and $|3 - 3| = 0$), so we print the number of chosen integers, 3, as our answer.

Sample Input 1

```
6
1 2 2 3 1 2
```

Sample Output 1

```
5
```

Explanation 1

We choose the following multiset of integers from the array: $\{1, 2, 2, 1, 2\}$. Each pair in the multiset has an absolute difference ≤ 1 (i.e., $|1 - 2| = 1$, $|1 - 1| = 0$, and $|2 - 2| = 0$), so we print the number of chosen integers, 5, as our answer.

Submissions: 1766
Max Score: 20
Difficulty: Easy

Rate This Challenge:
Thanks!
More

Current Buffer (saved locally, editable) C#

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

    static void Main(String[] args) {
        int n = Convert.ToInt32(Console.ReadLine());
           string[] a_temp = Console.ReadLine().Split(' ');
           int[] a = Array.ConvertAll(a_temp, e => int.Parse(e));

           //int[] a = { 4, 6, 5, 3, 3, 1 };
           //Console.WriteLine(pairs(arr));
           //int[] a = { 1, 2, 2, 3, 1, 2 };

           //int[] a = { 66, 66, 66, 66, 66, 66, 66, 66 };

           int max_len = 0;

           for (int i = 0; i < a.Length; i++)
           {
               int cont_actual = a.Count(e => e == a[i]);
               max_len = Math.Max(max_len, cont_actual);
               if (a.Contains(a[i] - 1))
               {
                   int cont_pre = a.Count(e => e == (a[i]-1));
                   max_len = Math.Max(max_len, cont_actual + cont_pre);
               }
               if (a.Contains(a[i] + 1))
               {
                   int cont_post = a.Count(e => e == (a[i] + 1));
                   max_len = Math.Max(max_len, cont_actual + cont_post);
               }
           }

           Console.WriteLine(max_len);
    }
}

```

Line: 23 Col: 36

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