TECHNICAL TEST – BACKEND DEVELOPER

Given an array of integers, find the longest subarray where the absolute difference between any two elements is less than or equal to 1

Example

a = [1,1,2,2,4,4,5,5,5]

There are two subarrays meeting the criterion:

[1,1,2,2] and [4,4,5,5,5] The maximum length subarray has elements.

Function Description

Complete the picking Numbers function in the editor below. Picking Numbers has the following parameter(s):

- int a[n]: an array of integers

Returns

- int: the length of the longest subarray that meets the criterion

Input Format

The first line contains a single integer n, the size of the a. The second line contains n space-separated integers, each a[i].

constraint

- $2 \leq n \leq 100$
- $0 < a[i] < 100$
- The answer will be ≥ 2.

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

Initial Script if you need

```
package main

import (
    "bufio"
    "fmt"
    "io"
    "os"
    "strconv"
    "strings"
)

func main() {
```

```
    reader := bufio.NewReaderSize(os.Stdin, 16 * 1024 * 1024)

    stdout, err := os.Create(os.Getenv("OUTPUT_PATH"))
    checkError(err)

    defer stdout.Close()

    writer := bufio.NewWriterSize(stdout, 16 * 1024 * 1024)

    nTemp, err := strconv.ParseInt(strings.TrimSpace(readLine(reader)), 10, 6
4)
    checkError(err)
    n := int32(nTemp)

    aTemp := strings.Split(strings.TrimSpace(readLine(reader)), " ")

    var a []int32

    for i := 0; i < int(n); i++ {
        aItemTemp, err := strconv.ParseInt(aTemp[i], 10, 64)
        checkError(err)
        aItem := int32(aItemTemp)
        a = append(a, aItem)
    }


    fmt.Fprintf(writer, "%d\n", result)

    writer.Flush()
}

func readLine(reader *bufio.Reader) string {
    str, _, err := reader.ReadLine()
```

```
    if err == io.EOF {
        return ""
    }

    return strings.TrimRight(string(str), "\r\n")
}

func checkError(err error) {
    if err != nil {
        panic(err)
    }
}
```