1955B - Progressive Square

Darvesh Aziz Mawla

28 April 2024

1 Problem

Problem Description : https://codeforces.com/problemset/problem/1955/B

2 Objective

For each test case, output "YES" in a separate line if a progressive square for the given n c and d can be constructed from the array elements a otherwise output "NO".

3 Solution

Since c¿0 and d¿0, the elements of the square increase starting from the top left corner. Therefore, a 1,1 is the minimum element in the square and consequently among the found elements. Given n, c, d, and the value of a 1,1, we can reconstruct the square. To verify that the given numbers in the input form the same square, we can simply sort both arrays of numbers and check for equality.

4 Code

```
#include<bits/stdc++.h>
using namespace std;

bool areEqual(int arr1[], int arr2[], int N, int M)
{
    unordered_map<int, int> mp;
    for (int i = 0; i < N; i++){
        mp[arr1[i]]++;
    }
    for (int i = 0; i < N; i++) {
        if (mp.find(arr2[i]) == mp.end())return false;
        if (mp[arr2[i]] == 0)return false;
        mp[arr2[i]]--;
    }
    return true;
}

int main(){
    int t;
    cin >> t;
    while(t-- > 0){
        int n,c,d,a11=2147483647;
        cin >> n >> c >> d;
        int arr[n*n] = {0};
        for(int i=0;i<n*n;i++){
            cin >> arr[i];
            a11 = min(a11,arr[i]);
        }
        int re[n*n] = {0};
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                re[(i*n) + j] = a11 + (i * c) + (j * d);
            }
        }
        int N = sizeof(arr) / sizeof(int);
        int M = sizeof(re) / sizeof(int);
        if (areEqual(arr, re, N, M)) cout << "YES" << endl;
        else cout << "NO" << endl;
    }
}
```