**Experiment 3.3**

**Competitive Coding Lab 10(Greedy, Branch and Bound)**

**Student Name: Sahul Kr. Parida UID: 20BCS4919**

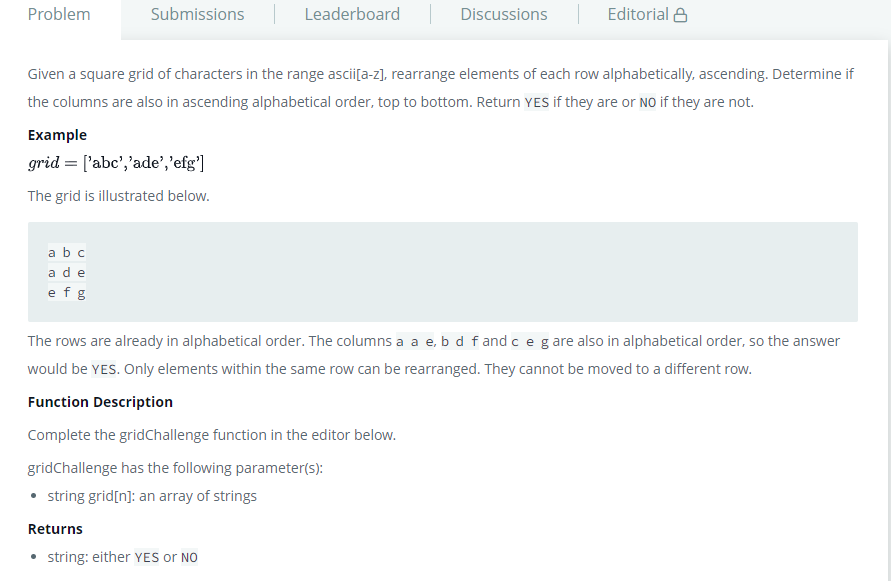
**Branch: CSE Section/Group: WM-904/B**

**Semester: 5th Date of Performance: 08/11/22**

**Subject Name: Competitive Coding(CC) Subject Code: 20CSP-314**

**PROBLEM STATEMENT 10.1: -**

<https://www.hackerrank.com/challenges/grid-challenge/problem?isFullScreen=false>



**SOLUTION:**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

    int T;

    cin>>T;

    while(T > 0)

    {

        int N;

        cin>>N;

        string \*matrix = new string[N];

        for(int i = 0; i < N; i++) {

            cin>>matrix[i];

            sort(matrix[i].begin(), matrix[i].end());

        }

        bool arrangeable = true;

        for(int i = 0; i < N; i++)

        {

            for(int j = 1; j < N; j++)

            {

                if(matrix[j][i] < matrix[j-1][i]) {

                    arrangeable = false;

                }

            }

            if(!arrangeable) {

                break;

            }

        }

        if(arrangeable)

{

            cout<<"YES\n";

        }

else

{

            cout<<"NO\n";

        }

        T--;

    }

    return 0;

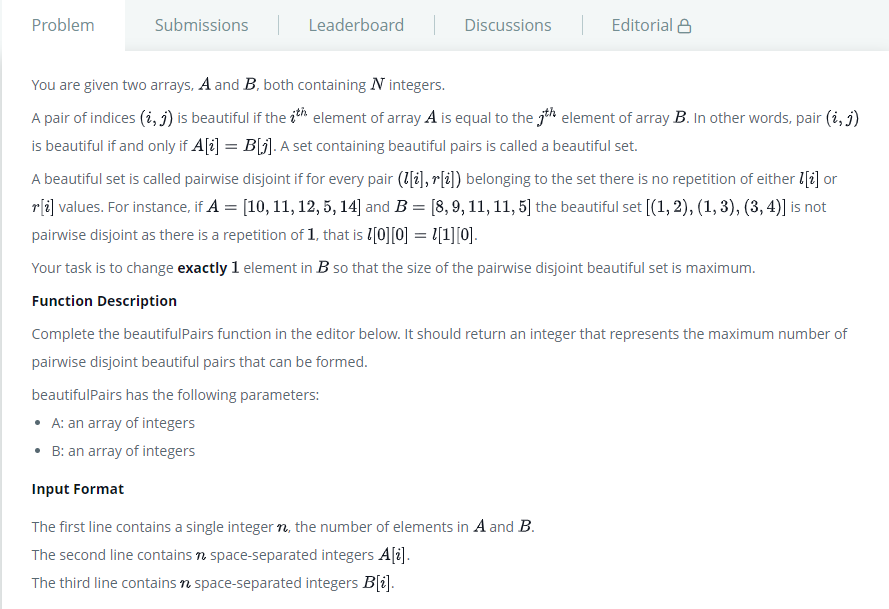
}

# TEST CASES:

# 

**PROBLEM STATEMENT 10.2: -**

<https://www.hackerrank.com/challenges/beautiful-pairs/problem?isFullScreen=false>



**SOLUTION:**

#include <cmath>

#include <cstdio>

#include <vector>

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

    int n;

    cin>>n;

    vector<int> a(1000, 0),b(1000, 0);

    for(int i = 0;i < n;i++)

    {

        int c;

        cin>>c;

        a[c]++;

    }

    for(int i = 0;i < n;i++)

    {

        int c;

        cin>>c;

        b[c]++;

    }

    int r = 0;

    for(int i = 1;i <= 1000;i++)

    {

        r += min(a[i], b[i]);

    }

    cout<<(r==n?n-1:r+1);

    return 0;

}

# TEST CASES:

# 