



## Experiment 10

### Competitive Coding Lab (Greedy, Branch and Bound)

**Student Name:** SANSKAR AGRAWAL

**UID:** 20BCS5914

**Branch:** CSE

**Section/Group:** MM-806/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>

Problem | Submissions | Leaderboard | Discussions | Editorial

Given a square grid of characters in the range `ascii[a-z]`, rearrange elements of each row alphabetically, ascending. Determine if the columns are also in ascending alphabetical order, top to bottom. Return YES if they are or NO if they are not.

#### Example

`grid = ['abc', 'ade', 'efg']`

The grid is illustrated below.

```
a b c
a d e
e f g
```

The rows are already in alphabetical order. The columns `a a e`, `b d f` and `c e g` are also in alphabetical order, so the answer would be YES. Only elements within the same row can be rearranged. They cannot be moved to a different row.

#### Function Description

Complete the `gridChallenge` function in the editor below.

`gridChallenge` has the following parameter(s):

- `string grid[n]`: an array of strings

#### Returns

- `string`: either YES or NO

### 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:


✓ Test case 0


✓ Test case 1 

✓ Test case 2 

✓ Test case 3 

✓ Test case 4 

✓ Test case 5 

✓ Test case 6 

Compiler Message

Success

Input (stdin)

Download

1	1
2	5
3	eabcd
4	fghij
5	olkmn
6	trpqs
7	xywuv

## PROBLEM STATEMENT 10.2: -

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

Problem

Submissions

Leaderboard

Discussions

Editorial 

You are given two arrays,  $A$  and  $B$ , both containing  $N$  integers.

A pair of indices  $(i, j)$  is beautiful if the  $i^{th}$  element of array  $A$  is equal to the  $j^{th}$  element of array  $B$ . In other words, pair  $(i, j)$  is beautiful if and only if  $A[i] = B[j]$ . A set containing beautiful pairs is called a beautiful set.

A beautiful set is called pairwise disjoint if for every pair  $(l[i], r[i])$  belonging to the set there is no repetition of either  $l[i]$  or  $r[i]$  values. For instance, if  $A = [10, 11, 12, 5, 14]$  and  $B = [8, 9, 11, 11, 5]$  the beautiful set  $[(1, 2), (1, 3), (3, 4)]$  is not pairwise disjoint as there is a repetition of  $1$ , that is  $l[0][0] = l[1][0]$ .

Your task is to change **exactly 1** element in  $B$  so that the size of the pairwise disjoint beautiful set is maximum.

### Function Description

Complete the beautifulPairs function in the editor below. It should return an integer that represents the maximum number of pairwise disjoint beautiful pairs that can be formed.

beautifulPairs has the following parameters:

- $A$ : an array of integers
- $B$ : an array of integers

### Input Format

The first line contains a single integer  $n$ , the number of elements in  $A$  and  $B$ .

The second line contains  $n$  space-separated integers  $A[i]$ .

The third line contains  $n$  space-separated integers  $B[i]$ .

## 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:**

✓ Test case 0

✓ Test case 1 

✓ Test case 2 

✓ Test case 3 

✓ Test case 4 

✓ Test case 5 

✓ Test case 6 

Compiler Message

Success

Input (stdin)

[Download](#)

1	4
2	1 2 3 4
3	1 2 3 3

Expected Output

[Download](#)

1	4
---	---