


[don\\_vanchos](#) | [Logout](#)
[HOME](#) [TOP](#) [CATALOG](#) [CONTESTS](#) [GYM](#) [PROBLEMSET](#) [GROUPS](#) [RATING](#) [EDU](#) [API](#) [CALENDAR](#) [HELP](#)
[FLAMESTORM](#) [BLOG](#) [TEAMS](#) [SUBMISSIONS](#) [GROUPS](#) [CONTESTS](#) [PROBLEMSETTING](#)

## flamestorm's blog

### Codeforces Round #806 (Div. 4) Editorial

By [flamestorm](#), 3 weeks ago, 

Thanks for participating!

1703A - YES or YES?

Idea: [flamestorm](#)

[Tutorial](#)

#### 1703A - YES or YES?

You should implement what is written in the statement. Here are three ways to do it:

- Check that the first character is Y or y, check that the second character is E or e, and check the third character is S or s.
- Make an array storing all acceptable strings (there are only 8), and loop and see if any of the strings match the input.
- Use some built-in function like `tolower()` in C++ to make the string all lowercase, and check if *s* is equal to **yes**.

The complexity is  $\mathcal{O}(1)$  per test case.

[Solution](#)

```
#include <bits/stdc++.h>

using namespace std;

const int MAX = 200007;
const int MOD = 1000000007;

void solve() {
    string s;
    cin >> s;
    if (s[0] != 'y' && s[0] != 'Y') {cout << "NO\n";}
    else if (s[1] != 'e' && s[1] != 'E') {cout << "NO\n";}
    else if (s[2] != 's' && s[2] != 'S') {cout << "NO\n";}
    else {cout << "YES\n";}
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int tt; cin >> tt; for (int i = 1; i <= tt; i++) {solve();}
    // solve();
}
```

1703B - ICPC Balloons

Idea: [flamestorm](#)

[Tutorial](#)

→ [Pay attention](#)

#### Before contest

[Codeforces Round #811 \(Div. 3\)](#)

05:23:34

[Register now »](#)

→ [don\\_vanchos](#)

 Rating: **1184**  
Contribution: **0**


don\_vanchos

- [Settings](#)
- [Blog](#)
- [Teams](#)
- [Submissions](#)
- [Favourites](#)
- [Talks](#)
- [Contests](#)

→ [Top rated](#)

#	User	Rating
1	<a href="#">tourist</a>	3771
2	<a href="#">jiangly</a>	3688
3	<a href="#">Um_nik</a>	3539
4	<a href="#">slime</a>	3498
5	<a href="#">djq_cpp</a>	3486
6	<a href="#">MiracleFaFa</a>	3466
7	<a href="#">ksun48</a>	3452
8	<a href="#">Radewoosh</a>	3406
9	<a href="#">greenheadstrange</a>	3393
10	<a href="#">xtqqwq</a>	3382

[Countries](#) | [Cities](#) | [Organizations](#)
[View all →](#)

→ [Top contributors](#)

#	User	Contrib.
1	<a href="#">awoo</a>	182
1	<a href="#">-is-this-fft-</a>	182
3	<a href="#">YouKn0wWho</a>	177
4	<a href="#">Um_nik</a>	175
5	<a href="#">Monogon</a>	172
5	<a href="#">dario2994</a>	172
7	<a href="#">antontrygubO_o</a>	168
7	<a href="#">maroonrk</a>	168
9	<a href="#">adamant</a>	167
10	<a href="#">errorgorn</a>	163

[View all →](#)

→ [Find user](#)

Handle:

## 1703B - ICPC Balloons

Let's keep an array  $a$  of booleans,  $a_i$  denoting whether or not some team has solved the  $i$ -th problem already. Now we can iterate through the string from left to right and keep a running total tot. If  $a_i$  is true (the  $i$ -th problem has already been solved), increase tot by 1; otherwise, increase tot by 2 and set  $a_i$  to true.

The time complexity is  $\mathcal{O}(n)$ .

Bonus: the answer is always  $n + \text{number of distinct characters in } s$ . Can you see why?

[Solution](#)

```
#include <bits/stdc++.h>

using namespace std;

const int MAX = 200007;
const int MOD = 1000000007;

void solve() {
    int n;
    cin >> n;
    string s;
    cin >> s;
    bool vis[26] = {};
    int res = 0;
    for (char c : s) {
        if (!vis[c - 'A']) {res += 2; vis[c - 'A'] = true;}
        else {res++;}
    }
    cout << res << '\n';
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int tt; cin >> tt; for (int i = 1; i <= tt; i++) {solve();}
    // solve();
}
```

## 1703C - Cypher

Idea: [mesanu](#)

[Tutorial](#)

## 1703C - Cypher

We will perform each move in reverse from the final sequence of the cypher.

- *down* move: it increases the  $i$ -th digit by 1. After applying the *up* move on 9, it becomes 0.
- *up* move (denoted by D): it decreases the  $i$ -th digit by 1. After applying the *down* move on 0, it becomes 9.

Now we just need to implement the two types of moves. The time complexity is  $\mathcal{O}(n + \sum a_i)$  per test case.

[Solution](#)

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

## Recent actions

[awoo](#) → [Educational Codeforces Round 132 Editorial](#) 🎉

[ScorpioDagger](#) → [Finally\\_Cyan! UPD: Now, BLUE!](#) 🎉

[Pranjalko](#) → [Why codeforces is not accepting this code when all the outputs are mathing correctly?](#) 🎉

[anonymous112233](#) → [HELP NEEDED: HARD RANGE QUERY PROBLEM](#) 🎉

[Vladosiya](#) → [Codeforces Round #811 \(Div. 3\)](#) 🎉

[WhyAsh5114](#) → [ZCO eligibility for students choosing non-conventional but valid education \(India\)](#) 🎉

[AAK](#) → [Indian ICPC 2022 Regionals – Discussion](#) 🎉

[Cirno\\_9baka](#) → [CodeTON Round 2 Editorial](#) 🎉

[QAQAutoMaton](#) → [IOI2022 China Team](#) 🎉

[AquaMoon](#) → [CodeTON Round 2](#) 🎉

[Ghosted](#) → [Maybe Codeforces needs a more complete anti cheating system](#) 🎉

[Qingyu](#) → [XXII Open Cup, Grand Prix of China \(EC-Finals\)](#) 🎉

[chokudai](#) → [AtCoder Beginner Contest 262 Announcement](#) 🎉

[hemant\\_thakur](#) → [Help needed in DP problem](#) 🎉

[Yubai](#) → [An Interesting combination problem](#) 🎉

[jli505](#) → [TeamsCode Summer 2022 Contest](#) 🎉

[Iftekhar\\_Hakim\\_K](#) → [Invitation to Replay of BUET Inter University Programming Contest 2022](#) 🎉

[marinyordanov](#) → [IATI 2018 \(ban?\)\\_\(Note that this is my new profile and therefore i am newbie :\)\)](#) 🎉

[scipianus](#) → [Codeforces Round #271 \(Div. 2\) Editorial](#) 🎉

[SyadouHayami](#) → [Ask for the solution of CF1704F](#) 🎉

[abhinav700](#) → [problem while declaring a 3d array](#) 🎉

[dsbdsb](#) → [fighting for rating 2200](#) 🎉

[RDFZzzx](#) → [A brief introduction of Segment Tree\(I\):Build tree and query without update](#) 🎉

[1900\\_mashups](#) → [Mashups for Aspiring Masters.](#) 🎉

[MikeMirzayanov](#) → [Rule about third-party code is changing](#) 🎉

[Detailed](#) →

```

void solve()
{
    int n;
    cin >> n;
    int a[n];
    for(int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
    for(int i = 0; i < n; i++)
    {
        int b;
        cin >> b;
        if(b == 0)
        {
            continue;
        }
        string now;
        cin >> now;
        for(int j = 0; j < b; j++)
        {
            if(now[j] == 'U'){a[i]--;}
            else if(now[j] == 'D'){a[i]++;}
            if(a[i] < 0){a[i]+=10;}
            if(a[i] > 9){a[i]-=10;}
        }
    }
    for(int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
}

int main(){
    int t;
    cin>> t;
    while(t--)
    {
        solve();
    }
    return 0;
}

```

### 1703D - Double Strings

Idea: [MikeMirzayanov](#)

[Tutorial](#)

## 1703D - Double Strings

Use some data structure that allows you to answer queries of the form: "does the string  $t$  appear in the array  $s_1, \dots, s_n$ ?" For example, in C++ you can use a `map<string, bool>`, while in Python you can use a dictionary `dict`.

Afterwards, for each string  $s$ , brute force all strings  $x$  and  $y$  such that  $s = x + y$ . There are at most 7 such strings, because  $s$  has length at most 8. Then check if both  $x$  and  $y$  appear in the array using your data structure.

The time complexity is  $\mathcal{O}(\ell n \log n)$  per test case, where  $\ell$  is the maximum length of an input string.

[Solution](#)

```
#include <bits/stdc++.h>

using namespace std;

const int MAX = 200007;
const int MOD = 1000000007;

void solve() {
    int n;
    cin >> n;
    string s[n];
    map<string, bool> mp;
    for (int i = 0; i < n; i++) {
        cin >> s[i];
        mp[s[i]] = true;
    }
    for (int i = 0; i < n; i++) {
        bool ok = false;
        for (int j = 1; j < s[i].length(); j++) {
            string pref = s[i].substr(0, j), suff = s[i].substr(j,
s[i].length() - j);
            if (mp[pref] && mp[suff]) {ok = true;}
        }
        cout << ok;
    }
    cout << '\n';
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int tt; cin >> tt; for (int i = 1; i <= tt; i++) {solve();}
    // solve();
}
```

## 1703E - Mirror Grid

Idea: [mesanu](#)[Tutorial](#)

## 1703E - Mirror Grid

Let's rotate the grid by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ , and mark all cells that map to each other under these rotations. For example, for  $4 \times 4$  and  $5 \times 5$  grids, mirror grid must have the following patterns, the same letters denoting equal values:

$$\begin{array}{cccc} a & b & c & a \\ c & d & d & b \\ b & d & d & c \\ a & c & b & a \end{array} \quad \begin{array}{ccccc} a & b & c & d & a \\ d & e & f & e & b \\ c & f & g & f & c \\ b & e & f & e & d \\ a & d & c & b & a \end{array}$$

In general, we can rotate the grid by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  and see which cells need to have equal values by seeing the positions which each cell maps to.

Now to solve the problem, we consider each equal value (each of the letters  $a, b, c, \dots$  in the above figures) independently, and consider the minimum number of moves to make them all 0 or all 1. The answer is the total across all values. See the implementation for better understanding.

The time complexity is  $\mathcal{O}(n^2)$  per testcase.

[Solution](#)

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

void solve()
{
    int n;
    cin >> n;
    int a[n][n];
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < n; j++)
        {
            char c;
            cin >> c;
            a[i][j] = c - '0';
        }
    }
    int ans = 0;
    for(int i = 0; i < (n+1)/2; i++)
    {
        for(int j = 0; j < n/2; j++)
        {
            int nowi = i, nowj = j;
            int oldnowj = nowj;
            int sum = a[nowi][nowj];
            nowj = n - nowi - 1;
            nowi = oldnowj;
            sum += a[nowi][nowj];
            oldnowj = nowj;
            nowj = n - nowi - 1;
            nowi = oldnowj;
            sum += a[nowi][nowj];
            oldnowj = nowj;
            nowj = n - nowi - 1;
            nowi = oldnowj;
            sum += a[nowi][nowj];
            ans += min(sum, 4 - sum);
        }
    }
    cout << ans << endl;
}

int main(){
    int t;
    cin >> t;
    while(t--)
    {
        solve();
    }
    return 0;
}
```

### 1703F - Yet Another Problem About Pairs Satisfying an Inequality

Idea: [flamestorm](#)

[Tutorial](#)

## 1703F - Yet Another Problem About Pairs Satisfying an Inequality

Call a pair *good* if it satisfies the condition. Let's split the inequality into three parts:

$a_i < i$ ,  $i < a_j$ ,  $a_j < j$ .

Note that if  $a_i \geq i$  for any  $i$ , then it can't be an element of a good pair, because it fails the first and third conditions. So we can throw out all elements of the array satisfying  $a_i \geq i$ .

For the remaining elements, the first and third inequalities are already satisfied, so we only have to count the number of pairs  $(i, j)$  with  $i < a_j$ . Let's iterate  $j$  through the array from the left to the right, and make a list storing all  $i$  that appear before  $j$ . Then for each  $j$ , count the number of  $i$  less than  $a_j$  by binary searching on the number of elements in the list less than  $a_j$ . Afterwards, add  $j$  to the end of the list.

Since we iterate from left to right, the list will always remain sorted (we insert the indices of elements, which are increasing from left to right), so the binary search will always work.

The time complexity is  $\mathcal{O}(n \log n)$ .

#### Solution

```
#include <bits/stdc++.h>

using namespace std;

const int MAX = 200007;
const int MOD = 1000000007;

void solve() {
    int n;
    cin >> n;
    int a[n + 1];
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    long long res = 0;
    vector<int> v;
    for (int i = 1; i <= n; i++) {
        if (a[i] >= i) {continue;}
        res += (long long)(lower_bound(v.begin(), v.end(), a[i]) -
v.begin());
        v.push_back(i);
    }
    cout << res << '\n';
}

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int tt; cin >> tt; for (int i = 1; i <= tt; i++) {solve();}
    // solve();
}
```

1703G - Good Key, Bad Key

Idea: [mesanu](#)

#### Tutorial

## 1703G - Good Key, Bad Key

We will prove it is always optimal to use good keys for a prefix then only use bad keys.

Consider we have used a bad key then a good key, by doing this we obtain

$\lfloor \frac{a_i}{2} \rfloor + \lfloor \frac{a_{i+1}}{2} \rfloor - k$  coins. If we switch and use a good key first, then a bad key then we obtain  $a_i + \lfloor \frac{a_{i+1}}{2} \rfloor - k$ , this number is clearly bigger so we will never encounter a bad key before a good key in an optimal solution, thus we will use a prefix of good keys then move on to using bad keys.

For every possible prefix of good keys we will calculate the coins we get at the end. We do this by maintaining a variable with the prefix sum where we use the good keys and then calculate what we will get from the chests where we use bad keys. Notice that because we halve all the chests when we use a bad key we only need to verify the next  $\log_2(10^9) \approx 30$  chests, all chests after it will go to 0 coins.

Final complexity:  $\mathcal{O}(n \log a_i)$ .

### Solution

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

void solve()
{
    int n, k;
    cin >> n >> k;
    int a[n];
    for(int i = 0; i < n; i++)
    {
        cin >> a[i];
    }
    long long ans = 0;
    long long sum = 0;
    for(int i = -1; i < n; i++)
    {
        long long now = sum;
        for(int j = i+1; j < min(n, i+32); j++)
        {
            int copy = a[j];
            copy>=j-i;
            now+=copy;
        }
        ans = max(ans, now);
        if(i+1 != n)
        {
            sum+=a[i+1]-k;
        }
    }
    cout << ans << endl;
}

int main(){
    int t;
    cin >> t;
    while(t--)
    {
        solve();
    }
    return 0;
}
```

 Tutorial of Codeforces Round #806 (Div. 4)

 +72 

 flamestorm

 3 weeks ago

 162



## Comments (162)

[Write comment?](#)



3 weeks ago, <#> | 

Quickly

→ [Reply](#)

poituk

 0 



3 weeks ago, <#> | 

Thanks for fast tutorial

→ [Reply](#)

[HideBeyondYou](#)

 +3 



3 weeks ago, # | ☆

+1

Amazing contest !! ..E is tricky one!

→ [Reply](#)[\\_xQc\\_](#)

3 weeks ago, # | ☆

0

It was exactly the same as rotating a 2d array using two for loops on leetcode.

→ [Reply](#)[goku20001](#)

3 weeks ago, # | ☆

0

That's why I Aced F instead of E.

→ [Reply](#)[dynamic\\_programming](#)

3 weeks ago, # | ☆

0

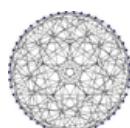
Yeah, I just used the same logic(leetcode 90 degree rotation) to create all four rotated configurations of the matrix and then checked min moves to make any cell equal in all four configurations.

→ [Reply](#)[ButterChickenn](#)

3 weeks ago, # | ☆

0

E is full implementation based

→ [Reply](#)[Snapper\\_001](#)

3 weeks ago, # | ☆

← Rev. 2

+60

F is solvable with prefix sums in  $O(n)$ .For each  $i$  count the number of elements, where  $j \leq i$  and  $a_j < j$  ( $\text{pref}_i$ ). And for each  $i$  where  $a_i < i$  just add  $\text{pref}_{a_i-1}$  to the answer.

Here is my solution 163953789.

→ [Reply](#)

3 weeks ago, # | ☆

+10

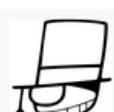
You beat me by 1 second, gonna say the same thing.

→ [Reply](#)[primeprogrammer2021](#)

3 weeks ago, # | ☆

← Rev. 6

-33

The comment is hidden because of too negative feedback,  
click here to view it[akash\\_musky](#)

3 weeks ago, # | ☆

0

can you please explain a bit more?

the prefix sum approach isn't really clear to me?

→ [Reply](#)[CoolKid\\_19](#)

3 weeks ago, # | ☆

+3

For every element we can store how many element we have encountered before it, that have  $v[i] < i$ 

```
int c = 0;
for(int i = 0; i < n; i++)
{
    cnt[i] = c;
    // 1 Based Indexing
```

```

    }  

}

```

Now for every element we can find whether we have it's value in range and whether we can find element so that we can satisfy the 2nd part in the condition i.e.  $i < v[i]$ , other conditions are already satisfied

```

int ans = 0;
for(int i = n - 1; i >= 0; i--)
{
    // We check for an element only when the element
    // itself follows the v[i] < i rule
    // AND
    // Only when its index is in our range 1 to n-1
    // (skipped 0 because there are no elements before it)
    // THEN
    // we find how many elements are before the
    // index equal to i since there is strict inequality
    if (i + 1 > v[i] && v[i] > 1)
        ans += cnt[v[i] - 1];
}

```

[My Submission](#)

→ [Reply](#)



2 weeks ago, <#> [^](#) | [☆](#)

[▲](#) **0** [▼](#)

woww great idea understood completely

→ [Reply](#)

guptaaman200229

4 days ago, <#> [^](#) | [☆](#)

← Rev. 2

[▲](#) **0** [▼](#)

the number of pairs  $i! = j$  of  $n$  elements is

$$C_n^2 = \frac{n(n-1)}{2} = \sum_{i=1}^{n-1} i$$



sky\_light

so you can keep a prefix sum of number elements that satisfied the condition  $i > a_i$

→ [Reply](#)



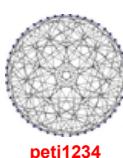
3 weeks ago, <#> [^](#) | [☆](#)

[▲](#) **0** [▼](#)

Can you plz tell me why your code is giving correct ans for the case when  $a[i]-1$  becomes  $-1$ . Beacause that should give a garbage value

→ [Reply](#)

goku20001



3 weeks ago, <#> [^](#) | [☆](#)

[▲](#) **0** [▼](#)

My bad, thanks for noticing. Probably it is 0 in this case.

Now the code is correct.

→ [Reply](#)



3 weeks ago, <#> [^](#) | [☆](#)

← Rev. 6

[▲](#) **0** [▼](#)

i've solved it Using ordered Multiset:



after Getting a value which is lesser than its index just check how many values are present in the set which is lesser than this...

and after this just insert the index of the value ( not the value! to satisfy " $"ai < aj < j$  this"")

→ [Reply](#)

[▲](#) **-10** [▼](#)

3 weeks ago, <#> [^](#) | [☆](#)

Can G be solved by DP ?

→ [Reply](#)

Hamroz\_JONN



suraj10

3 weeks ago, # ⌂ | ☆

0

Yes I solved G using DP. I used the fact that applying the bad operation 30 times would make all elements 0 since  $\log(1e9)$  is approximately 30.

Here's my solution:

<https://codeforces.com/contest/1703/submission/163933460>→ [Reply](#)

Munnoo

3 weeks ago, # ⌂ | ☆

0

Yes, I made a Recursion + Memoization solution, using similar concept. Check it here : [Recursive Solution](#)

→ [Reply](#)

mdvorak

3 weeks ago, # ⌂ | ☆

0

Just wanted to comment the exact same thing. My solution

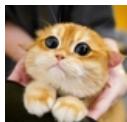
→ [Reply](#)

minh20cm

2 weeks ago, # ⌂ | ☆

0

:))

→ [Reply](#)

BrxDeputat

3 weeks ago, # | ☆

+3

I will say that it was a good contest. Thanks for the quick tutorial

→ [Reply](#)

ProgramMastergg

3 weeks ago, # | ☆

0

I'll just say it. This contest is the best for me .

→ [Reply](#)

jha\_rishi

3 weeks ago, # | ☆

0

D and F are good problems.

→ [Reply](#)

Silverbridge

3 weeks ago, # | ☆

0

Hey! Can anyone clarify why my solution is showing TLE on Test Case 10 (Problem D), my approach is similar as that mentioned in the Editorial. Thanks!

[163871330](#)→ [Reply](#)

hydroshiba

3 weeks ago, # ⌂ | ☆

← Rev. 2

+6

Your problem lies here I suppose:

Spoiler

When you wrote `ans = ans + '1'` what the code did was calculate `ans + '1'` then assign that resulted string to `ans`, which made this operation  $O(N)$ . Use `ans += '1'` next time.

Bonus: Here's your very same code accepted with those changes.

→ [Reply](#)

Silverbridge

3 weeks ago, # ⌂ | ☆

0

Thanks!

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

← Rev. 2

0

You just need to speed up the I/O. Just add this line (before defining t int your main function).

`ios::sync_with_stdio(false); cin.tie(nullptr);` After that

YesPy

~~↳ sync\_with\_stdio(false), cin.tie(nullptr);~~

tabulate the code correctly and you'll get AC.

[→ Reply](#)

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

No it didn't

hydroshiba



3 weeks ago, # ⌂ | ☆

▲ 0 ▼

But your tabulation is not correct.

[→ Reply](#)

YesPy



3 weeks ago, # ⌂ | ☆

▲ 0 ▼

But who asked.

[→ Reply](#)

chromate00

3 weeks ago, # ⌂ | ☆

← Rev. 2

▲ +5 ▼

I think problem F can be done in  $O(n)$  time.As mentioned, we only need to consider indices  $i$  such that  $a_i < i$ . Call such indices "good". We can make a prefix array such that the  $i$ th element is the number of good indices less than or equal to  $i$ .Now consider all possible pairs  $(i, j)$  that work for a fixed good index  $j$ . The number of possible values for  $i$  is the number of good indices less than  $a_j$ , which is stored within the prefix array. The answer can then be found by summing across all good indices  $j$ .

Bryce3D

The prefix array can be created in 1 pass, so that takes  $O(n)$  time. The summing across all good indices can also be done in 1 pass, so this is  $O(n)$  time again.

Code for this can be found in my submission

Edit: Apparently got sniped lmao

[→ Reply](#)

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Can you plz tell me why your code is giving correct ans for the case when  $a[i]-1$  becomes  $-1$ . Beacause that should give a garbage value[→ Reply](#)

goku20001



3 weeks ago, # ⌂ | ☆

▲ 0 ▼

I specifically excluded it in the if condition to handle that bug

 $\text{if } a[i] < i+1 \text{ and } a[i] > 0:$ [→ Reply](#)

Bryce3D

divyansh\_sahu



3 weeks ago, # ⌂ | ☆

← Rev. 2

▲ 0 ▼

Had a similar idea but instead of using a prefix array, for all indices  $i$  where  $a_i < i$ , I wanted to count all the indices  $j$  ahead of  $i$  where  $a_j < j$ . This however, would not work for cases where  $a_j <= a_i$ .Then I left the question thinking I needed to know segment trees etc to solve it in  $O(n \log n)$  :(

Solved it after the contest using prefix array though.

[→ Reply](#)

some\_randomguy



3 weeks ago, # ⌂ | ☆

▲ 0 ▼

I have done the same. What do you mean by "sniped" ?

[→ Reply](#)



3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Someone else said a similar thing while I was typing my message

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

← Rev. 2 ▲ 0 ▼

Anyone could tell me what's wrong with this approach?

→ [Reply](#)**kalavalarevanth**

3 weeks ago, # ⌂ | ☆

▲ +8 ▼

Problem G was a good one.

→ [Reply](#)**abhi\_ram**

2 weeks ago, # ⌂ | ☆

▲ 0 ▼

Yes

→ [Reply](#)**white Devil\_403**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

For the problem D we can use `unordered_set<string> cache` to store the strings. Then for each string, bruteforce divide the strings in two and check whether both of these strings exist in the `cache` or not. Simple

→ [Reply](#)**quater\_nion**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

to be honest though, you should be very cautious on using hash-based data structures in codeforces. Some people are masters of hacking and can easily find hundreds, even thousands of hash collisions.

→ [Reply](#)**chromate00**

3 weeks ago, # ⌂ | ☆

▲ +1 ▼

Thats true... I think one way we can avoid is by making 2 or three hashes differently and storing in each of them. That way the chance of getting same hashes reduces significantly

→ [Reply](#)**quater\_nion**

3 weeks ago, # ⌂ | ☆

← Rev. 2 ▲ -15 ▼

This code of mine for question D(double strings) is giving TLE on test 3 and I don't know why. Please help me where is the problem?

**Royal\_Drax**

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int

char check_string(ll k,vector<string> &v){
    if(v[k].size() == 1) return '0';
    for(ll i=0;i<v.size();i++){
        if(i == k) continue;
        for(ll j=0;j<v.size();j++){
            if(j == k) continue;
            if(v[i]+v[j] == v[k]) return '1';
        }
    }
    return '0';
}

int main(){
    ll t;
    cin>>t;
    while(t--){
        cout<<check_string(t,vec);
    }
}
```

```

    ll n;
    cin>>n;
    vector<string> v(n);
    for(ll i=0;i<n;i++){
        cin>>v[i];
    }
    string s;
    for(ll i=0;i<n;i++){
        s += check_string(i,v);
    }
    cout<<s<<endl;
}
return 0;
}
→ Reply.

```



3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Your solution is  $O(n^2)$ .→ [Reply](#).**Lanceloia**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

use markdown correctly plz

→ [Reply](#).**QingFy**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Can F be solved by Fenwick tree??

→ [Reply](#).**Kousei\_Arima**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Yes. 163916062

→ [Reply](#).**YesPy**

2 weeks ago, # ⌂ | ☆

▲ 0 ▼

Yes

→ [Reply](#).**white Devil\_403**

3 weeks ago, # ⌂ | ☆

← Rev. 3

▲ +7 ▼

any solution which can count pairs where  $i < a_j$  where  $a_i < i, a_j < j$  in all  $i$  and  $j$  is a valid solution. therefore Segment Trees and Fenwick Trees are valid solutions, so are prefix sums.→ [Reply](#).**chromate00**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Yes. I have done it using segment tree. here is my solution 163925757

→ [Reply](#).**Tirthsuthar**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Honestly, F was too easy with Segment Trees (or Fenwick Trees, those two have identical purposes). I think any person with some experience of common segment tree problems would easily solve it.

→ [Reply](#).**chromate00**

3 weeks ago, # ⌂ | ☆

▲ +1 ▼

BS enters the room\*

Prefix Sum enters next and slap both on the face\*

→ [Reply](#).**Platanito\_Frito**



3 weeks ago, # ⌂ | ☆

+2

it can be easily solved with pbds with a few lines of code

→ [Reply](#)**MuheshKumar**

3 weeks ago, # ⌂ | ☆

+1

yes, very easy code.

→ [Reply](#)**tgp07**

3 weeks ago, # ⌂ | ☆

+14

Let's see in the hacking phase if people learned to use the map instead of the unordered\_map. **beethoven97** I summon you!→ [Reply](#)**Platanito\_Frito**

3 weeks ago, # ⌂ | ☆

← Rev. 2 0

why this solution 163935126 giving TLE ? Anybody please help

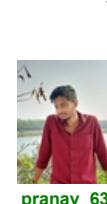
→ [Reply](#)**pranav\_63**

3 weeks ago, # ⌂ | ☆

0

because memset function fills the DP using its size --&gt; O(N), here it is N = 1e5 so your dp gets filled for all test cases which can be upto 1e4, so ~1e9 operations per second are performed.

use this

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

0

Thank you

→ [Reply](#)**pranav\_63**

3 weeks ago, # ⌂ | ☆

0

you are initializing the dp array in each test case. There can be up to 1e4 test cases. Even if n is 1 in all test cases, you're initializing an array of size 3e6 in each test case. In the worst case, you'll be doing 3e10 operations which is way out of the problem constraints. To avoid this, you can declare a dynamic array(vector) and resize it in each test case according to the value of n. Because we're guaranteed that the sum of n across all test cases doesn't exceed 1e5

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

0

Thank you.

→ [Reply](#)**pranav\_63**

3 weeks ago, # ⌂ | ☆

0

You can try something like this. 163938289

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

0

Thanks for this amazing contest and fast tutorial.

→ [Reply](#)**budushiyprogrammist**

3 weeks ago, # ⌂ | ☆

← Rev. 2 +3

why my dp solution doesn't work for G

<https://codeforces.com/contest/1703/submission/163929073>→ [Reply](#)**LakshaySodhi29**

3 weeks ago, # ⌂ | ☆

0

For the fast case





Sidharth\_71

1 0 0 1

1 0 0 1

1 0 0 1

→ [Reply](#)

glitchsin

3 weeks ago, # ⌂ | ☆

0

4 moves are required. U have to make either top 2 zeroes and bottom 2 zeroes to one or u can either make left two ones and right two ones to zero.

→ [Reply](#)

Amer\_Mahfudh

3 weeks ago, # ⌂ | ☆

0

Amazing contest , I've gained four ideas from issues I thank those who wrote those for issues and I hope that the contests div4 will increase in future

→ [Reply](#)

HNOONa

3 weeks ago, # ⌂ | ☆

0

i tried to solve F with multiset (c++) along with lower\_bound, distance methods here: [link](#)

but i get TLE, why?

→ [Reply](#)

filippodigra

3 weeks ago, # ⌂ | ☆

0

distance has O(N) complexity for set e multiset (N is the size of the set)

→ [Reply](#)

HNOONa

3 weeks ago, # ⌂ | ☆

← Rev. 2

0

is there a way around this to get O(1) distance operation?

→ [Reply](#)

filippodigra

3 weeks ago, # ⌂ | ☆

0

distance has O(1) complexity only with random access data structures such as vectors. check the complexity section:

<https://mcplusplus.com/reference/iterator/distance/>→ [Reply](#)

HNOONa

3 weeks ago, # ⌂ | ☆

0

ok thanks

never doing that again

→ [Reply](#)

Tanish69

3 weeks ago, # ⌂ | ☆ 0

I implemented it using 2 sets.  
Check my submission.

→ [Reply](#)

\_\_HACKER\_\_

3 weeks ago, # ⌂ | ☆

← Rev. 2

0

<https://ideone.com/diXeOD> (My code) Problem F using Ordered Set. Simple Code.. Hope it will help!

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

← Rev. 2

0

okay, so order\_of\_key is like lower\_bound? without iterator part?

→ [Reply](#)



HNOONa



pro\_chinkkkk

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

"the number of items in a set that are strictly smaller than our item". you can see here  
<https://codeforces.com/blog/entry/11080>

[→ Reply](#)

HNOONa



\_\_HACKER\_\_

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

← Rev. 2

▲ 0 ▼

It returns how many values are strictly lesser than this in order of  $\log(n)$  time

[→ Reply](#)

Milsu

3 weeks ago, # | ☆

▲ 0 ▼

So amazing

[→ Reply](#)

lemon\_fresh

3 weeks ago, # | ☆

▲ 0 ▼

Can anyone tell why using vector in this solution for problem E gives wrong answer? Replacing vector with plain array makes it accepted. Submission link

[→ Reply](#)

sumitprajapati

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Look at the case when  $x = 0$ , you are accessing  $\text{sum}[x-1]$  or  $\text{sum}[-1]$ [→ Reply](#)

lemon\_fresh

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

You're right. I guess it was luck that it got accepted with c array.

[→ Reply](#)

tushar\_kumar



adityagamer

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

someone hacked my D (163901816) can anyone provide me test case ?

[→ Reply](#)

Literally\_Me

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

$\text{ans} = \text{ans} + '1'$  makes copy of ans then adds '1' you should use  $\text{ans} += '1'$

[→ Reply](#)

Royal\_Drax

3 weeks ago, # | ☆

▲ 0 ▼

Why 163895211 is giving TLE?

[→ Reply](#)



3 weeks ago, # | ☆  
DP solution for G. Lang: PyPy3-64

← Rev. 4 ▲ 0 ▼

---

```
t = int(input())

for _ in range(t):
    n, k = map(int, input().split())
    arr, = map(int, input().split())

    MAX_HALF_CNT, NINF = 1, -2**63
    max_coins = max(arr)
    while max_coins >> MAX_HALF_CNT:
        MAX_HALF_CNT += 1
    MAX_HALF_CNT += 1

    dp = [[NINF] * MAX_HALF_CNT for _ in range(n)]
    # dp[i][half_cnt] 打开第i个抽屉后所能得到的最大硬币数, 其中使用了half_cnt次坏钥匙,

    dp[0][0] = arr[0] - k # 用好钥匙, 减k
    dp[0][1] = (arr[0] >> 1) # 用坏钥匙, 减半
    # print(dp)
    for i in range(n-1):
        for h in range(MAX_HALF_CNT):
            dp[i+1][h] = max(dp[i+1][h], dp[i][h] + (arr[i+1] >> h) - k)
            if h + 1 < MAX_HALF_CNT:
                dp[i+1][h+1] = max(dp[i+1][h+1], dp[i][h] + (arr[i+1] >> (h + 1)))
            else:
                # 哪都没了
                dp[i+1][h] = max(dp[i+1][h], dp[i][h] + 0)

    print(max(dp[n-1]))
```

---

→ [Reply](#)



3 weeks ago, # | ☆  
The idea for solving Problem F was very nice  
→ [Reply](#)

second\_saturday



3 weeks ago, # | ☆  
Could help me G problem? i use dp to slove it ,but i cant fine my problem.thank u very much

\_zjb

problem G  
TAT...my english is poor ,but i try to share my idea

j:the number of good key

$pw(2,x)=2^{**x}$

when we open the i th box ,if we use j good keys so we use  $(i-j)$  bad keys.but  
 $pw(2,33)>1e9$ ,so i think the number of good key is lower than 33.

if the number of bad key is upper than 33,we can see it as  $1e14$ .

so when we open some box,if we use good keys ,we got  $a[i]/(pw[(i-j)] - m)$  cost

and we use bad keys ,we got  $a[i]/(pw[(i-j)] - m)$

so DP[i][j] is mean open i boxes use j good keys.

$Dp[i][j]=Dp[i-1][j]+$  bad keys or  $DP[i][j]=DP[i-1][j-1] +$  good keys

so what's wrong with me TAT

[go what's wrong with me TAT](#)[→ Reply](#)3 weeks ago, <#> [^](#) | [☆](#)[+2](#)

Suppose that **n** and **k** are very large And the optimal operations are always use bad keys. In this case obviously we use more than 32 bad keys but your solution don't consider this and gives the wrong answer.

**Parsa\_Sabzei**

To fix it : for calculating answer iterate over all dp values and get maximum instead of only **dp[n - 1]** (because if we do this we consider above case):

```
for(int i = 0; i < n; i++)
    for(int j = 0; j < 32; j++)
        ans = max(ans, dp[i][j]);
```

My submission : 163915709

[→ Reply](#)3 weeks ago, <#> [^](#) | [☆](#)[0](#)I am getting TLE on second TC, can you please help [LINK](#)

Thanks.

[→ Reply](#)**hardik2001**3 weeks ago, <#> [^](#) | [☆](#)[0](#)

I think you need let done be 31

[→ Reply](#)**\_zjb**3 weeks ago, <#> [^](#) | [☆](#)[0](#)

thank you, I know

[→ Reply](#)**\_zjb****pr7nce**3 weeks ago, <#> | [☆](#)[← Rev. 3](#) [0](#)

Can anyone help me find why I'm getting TLE on 5th case and how to further optimize this code?

```
#include<bits/stdc++.h>
using namespace std;
vector<vector<long long>> dp;
long long rec(vector<long long>& arr, long long k, long long factor,
long long i){
    if(i>=arr.size())
        return 0;
    if(dp[i][factor]!=-1)
        return dp[i][factor];
    long long temp=-k+arr[i]/pow(2, factor)+rec(arr, k, factor, i+1);
    long long t=arr[i]/pow(2, factor+1)+rec(arr, k, factor+1, i+1);
    if(temp>t)
        return dp[i][factor]=temp;
    return dp[i][factor]=t;
}
int main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    long long t;
    cin>>t;
    while(t--){
        dp.clear();
        long long n, k;
        cin>>n>>k;
        vector<long long> arr(n);
        for(long long i=0;i<n;i++)
```

[Answer](#)

```

    cout<<arr[i];
    dp.resize(n+1, vector<long long>(n+2, -1));
    cout<<rec(arr, k, 0, 0)<<endl;
}
}
→ Reply

```

**hardik2001**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

[link](#) There you go, Its working now

I changed your dp size n , 32 and early pruning in base case

→ [Reply](#)

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Thanks a lot man.

→ [Reply](#)**pr7nce****Dasher001**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

For which test case it fails.. [My solution](#)→ [Reply](#)**hggf**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

For first 30 minutes i think in E we need to make matrix by add 1. Later i eays accepted this Problem

→ [Reply](#)**Silverbridge**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Hey! Can someone please clarify why my Dp solution for Problem G is showing TLE on Test Case 3? Thanks!

163941930

→ [Reply](#)**altjustforcomment**

3 weeks ago, # ⌂ | ☆

← Rev. 2 ▲ 0 ▼

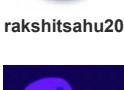
G is quite hard, other problems are great. Hope the rating will change soon so well-perform participants could get to the next rank!

→ [Reply](#)**rakshitsahu20**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

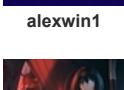
I did a,b,c,d,e,f with 0 wrong submission can i expect pupil ... Now

→ [Reply](#)**alexwin1**

3 weeks ago, # ⌂ | ☆

▲ 0 ▼

Thanks for the fast editorial

→ [Reply](#)**Anton\_Vdovin**

3 weeks ago, # ⌂ | ☆

← Rev. 2 ▲ 0 ▼

Another F approach. We can create **p** where **p[i] = (a[i] < i)**, after that create a prefix sum on that array and iterate over **a** and do the following:

- check if **p[i] = 1**, otherwise continue
- check if **a[i] >= 2**, otherwise continue
- add **prefsum[a[i] — 1]** to the answer

So why do we need **a[i] >= 2** statement? If **a[i] < 2**, then there is no way to satisfy this inequality. **a[i] >= 0, i >= 1**, therefore **a[i] < i < 1 < j** can't be satisfied. Check out my implementation if you are interested [164005949](#). What do you think about this one?

P.S Well, just find out some guy already told about the same approach...

→ [Reply](#)

→ [Reply](#)

3 weeks ago, # | ☆

▲ 0 ▼

Simple recursive DP solution for G here.

→ [Reply](#)

3 weeks ago, # | ☆

▲ +5 ▼

`map<string, bool>` ? Why not just use a `set<string>` ?→ [Reply](#)

ivan100sic

3 weeks ago, # | ☆

← Rev. 2 ▲ 0 ▼

Can anyone help to reduce time complexity of G



MoSalah10

```
#include <bits/stdc++.h>
using namespace std;
long long arr[32];
long long rec(long long i, long long n, long long p, vector<vector<long long>>& dp, long long k, vector<vector<long long>> & v, long long v[], long long arr[])
{
    if(i==n || p>30) return 0;
    if(dp[i][p]==-1) return dp[i][p];
    int x = v[i]/arr[p];
    return dp[i][p] = max(x-k+rec(i+1,n,p,v,k,dp),x/2+rec(i+1,n,p+1,v,k,dp));
}
int main()
{
    long long t;cin >> t;
    arr[0] = 1;
    for(long long i=1;i<32;i++) arr[i] = 2*arr[i-1];
    while(t--){
        long long n,a,ans=0,k; cin >> n >> k;
        vector<long long> v(n,0);
        vector<vector<long long>> dp(n,vector<long long>(32,-1));
        for(long long i=0;i<n;i++) cin >> v[i];
        cout << rec(0,n,0,v,k,dp) << endl;
    }
    return 0;
}
```

→ [Reply](#)

3 weeks ago, # | ☆

▲ 0 ▼

Can someone tell me why this dp solution is getting WA, doesn't make any sense to me? <https://codeforces.com/contest/1703/submission/163945776>→ [Reply](#)

Nidz05



3 weeks ago, # ▲ | ☆

▲ 0 ▼

You also have to update your answer while calculating dp so as to counter cases when you use more than 30 bad moves. One such example will be, 60 chests containing 2 coins and lets say k = 1e9.

→ [Reply](#)

epsilon\_573



3 weeks ago, # ▲ | ☆

▲ 0 ▼

You are saying that i have to take the max of the whole matrix?

→ [Reply](#)

Nidz05



3 weeks ago, # ▲ | ☆

▲ 0 ▼

Not really the whole matrix, max of  $dp[i][30]$  over all chests  $i$  + your current logic will be sufficient.→ [Reply](#)

epsilon\_573



Nidz05

3 weeks ago, # ⌈ | ☆

0

Okay thanks, i fixed it

[→ Reply](#)

3 weeks ago, # ⌈ | ☆

← Rev. 2 0

not max { dp[n][i] }, try max{ dp[i][j] } because maybe after dp[i][j] all cost of box if 0 so maybe this i is not n.

[→ Reply](#)

epsilon\_573

3 weeks ago, # ⌈ | ☆

0

Video Solutions of complete problemset.

[→ Reply](#)

Yousef\_Sameh



ishu\_rc

3 weeks ago, # ⌈ | ☆

0

did ur rating update? i am newbie but rating not updated

[→ Reply](#)

Yousef\_Sameh



lis05

3 weeks ago, # ⌈ | ☆

0

F can be solved in linear time. submission

[→ Reply](#)

maddler

3 weeks ago, # ⌈ | ☆

0

What is the rating range for problems F and G?

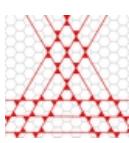
[→ Reply](#)

ishu\_rc

3 weeks ago, # ⌈ | ☆

0

hey my rating is not updated even I am a newbie and gave 8-9 rated contests before

[→ Reply](#)

nonrice

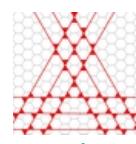
3 weeks ago, # ⌈ | ☆

0

Ratings did not seem to update. Anyone else having this problem?

[→ Reply](#)

rakshitsahu20



3 weeks ago, # ⌈ | ☆

0

Just checked again. Ratings are updated now.

[→ Reply](#)



RahulAhuja2901

3 weeks ago, # ⌂ | ⭐

0

Can anyone please tell why am I getting Memory Limit Exceeded in Problem E? I am just using a 2D Array. Link to my submission :-  
<https://codeforces.com/contest/1703/submission/163908463>

[→ Reply](#)

3 weeks ago, # ⌂ | ⭐

0

Hello sir, There are two layers of recurrence in your code, so your code's time complexity is  $O(n^2)$ , and when  $n = 200000$ , you will get TLE.

I solved this problem with set, and my time complexity is  $O(n \log n)$ .

This is my code:

```
# include <bits/stdc++.h>
#define ll long long
#define re register
#define il inline
using namespace std;
string s[100010];
int main()
{
    int t;
    scanf("%d", &t);
    while (t--)
    {
        int n; bool f = 0;
        scanf("%d", &n);
        set<string> str;
        for (int i = 1; i <= n; ++i)
        {
            cin >> s[i];
            str.insert(s[i]);
        }
        for (int i = 1; i <= n; ++i)
        {
            f = 0;
            for (int j = 1; j < s[i].size(); ++j)
            {
                string a = s[i].substr(0, j), b = s[i].substr(j, s[i].size());
                if (str.find(a) != str.end() && str.find(b) != str.end())
                {
                    printf("1");
                    f = 1;
                    break;
                }
            }
            if (f == 0) printf("0");
        }
        printf("\n");
    }
    return 0;
}
```



zhaoxi\_zheng

Although there are two layers of recurrence in my code, but  $s[i].size \leq 8$ , so my code doesn't go TLE.

[→ Reply](#)

3 weeks ago, # ⌂ | ⭐

0

Hey, Thanks for the reply but I am asking about Problem E, Mirror Grid. The solution that you have explained is Problem

D Double Strings

RahulAhuja2901

D, Double Damage.

[→ Reply](#)2 weeks ago, <#> [^](#) | [☆](#)[▲ 0](#)

Sorry, it's my mistake. Let me show you my code of problem E.

zhaoxi\_zheng

```
# include <bits/stdc++.h>
#define ll long long
#define re register
#define il inline
using namespace std;
char g[110][110];
int f[110][110];
int cnt[5];
int main()
{
//    freopen("output.out", "w", stdout);
    int t;
    scanf("%d", &t);
    while (t--)
    {
        int n, ans = 0;
        scanf("%d", &n);
        for (int i = 1; i <= n; ++i)
        {
            string s;
            cin >> s;
            for (int j = 1; j <=
n; ++j)
                g[i][j] =
int(s[j - 1] - '0');
            }
            int len = (n + 1) >> 1;
            for (int i = 1; i <= len;
++i)
            {
                int k;
                if (n % 2) k = len -
1;
                else k = len;
                for (int j = 1; j <=
k; ++j)
                {
                    cnt[0] =
cnt[1] = 0;
                    int x = n -
i + 1, y = n - j + 1;
                    cnt[g[i]
[j]]++;
                    cnt[g[j]
[x]]++;
                    cnt[g[y]
[i]]++;
                    cnt[g[x]
[y]]++;
                    //    printf("%d
%d %d\n", i, j, cnt[0], cnt[1]);
                    ans += min(4
- cnt[0], cnt[0]);
                }
            }
            printf("%d\n", ans);
        }
return 0;
}
```

}

[→ Reply](#)

8 days ago, # ▲ | ☆

You are creating new 2D array every test case.

[→ Reply](#)

▲ 0 ▼



RahulAhuja2901

8 days ago, # ▲ | ☆

Thank you so much for replying. Can you tell what should be approach for this problem?

[→ Reply](#)

▲ 0 ▼



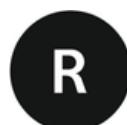
vito\_t

8 days ago, # ▲ | ☆

To be honest I didn't solve this problem and now I don't have time now because the contest starts in 2 minutes.

To fix your problem (MLE) you can create 2D array before first test case. (before `while (t > 0)` line)[→ Reply](#)

▲ 0 ▼



RahulAhuja2901

8 days ago, # ▲ | ☆

Because in the editorial they are also creating a 2D array for each test case.

[→ Reply](#)

▲ 0 ▼



vito\_t

8 days ago, # ▲ | ☆

Then I don't know. Maybe I was wrong.

[→ Reply](#)

▲ 0 ▼



np\_incomplete

3 weeks ago, # | ☆

← Rev. 2

▲ 0 ▼

Loved the round! Wish I could solve G. Thanks to the setters and testers.

[→ Reply](#)

np\_incomplete

3 weeks ago, # | ☆

▲ +3 ▼

I solved F in O(n) Here is my solution: 163897273

[→ Reply](#)

jcnp

3 weeks ago, # | ☆

▲ 0 ▼

Why also use map&lt;string, bool&gt; mp in D, but i Time limit exceeded.

[→ Reply](#)

utku\_1234

3 weeks ago, # | ☆

← Rev. 2

▲ 0 ▼

i did question f in O(n logn) but still it is showing me TLE in one test case can anyone help me out please .... i would be grateful my sol 164096201

[→ Reply](#)

NafisAlam

2 weeks ago, # ▲ | ☆

▲ 0 ▼

you have to pass the vector by reference in the find\_pos function

[→ Reply](#)

3 weeks ago, # | ☆

▲ 0 ▼

C no Problem is quite interesting! in my opinion

[→ Reply](#)

\_INFINITY\_LOOP\_

3 weeks ago, # | ☆  
Good solution for problem G!  
→ [Reply](#)

v7fgg

▲ 0 ▼



3 weeks ago, # | ☆  
For question good keys bad keys why is recursion + memo wrong  
my submission 164141664  
→ [Reply](#)

▲ 0 ▼



3 weeks ago, # ▲ | ☆ ← Rev. 2 ▲ 0 ▼  
figured out the solution. made a silly mistake. Ignore the above comment  
→ [Reply](#)

ajblahblah



3 weeks ago, # ▲ | ☆ ← Rev. 2 ▲ 0 ▼  
I did the same recursion + memo. check here: [Recursion+Memo](#)  
→ [Reply](#)

Munnoo



2 weeks ago, # | ☆  
Note that, in problem G, don't use memset all the times, or you will get TLE on test 2 (I had tried it), because memset has a time complexity as  $O(n \times \log a_i)$ , if  $t$  is very big, the total time complexity will be too large to solve this problem.

zhaoxi\_zheng

▲ 0 ▼



I hope that this note will help you!  
→ [Reply](#)

2 weeks ago, # | ☆  
I am getting runtime error in G. Please help me out. Here is my [submission](#)  
→ [Reply](#)

lol\_py

▲ 0 ▼



2 weeks ago, # | ☆  
Can someone suggest more problems similar to G?(like DP states like this)  
→ [Reply](#)

wackyindian

▲ 0 ▼



2 weeks ago, # | ☆  
good round  
→ [Reply](#)

antsparrow5

▲ 0 ▼



2 weeks ago, # | ☆  
good  
→ [Reply](#)

Ahmed\_Nageh\_Abbas

▲ 0 ▼



2 weeks ago, # | ☆  
Good Tutorial!  
→ [Reply](#)

Arthur\_Weng

▲ 0 ▼



12 days ago, # | ☆  
Can someone please explain to me what is wrong with my solution for F or provide a counter example? I probably messed up my binary search implementation but cannot figure it out.  
<https://codeforces.com/contest/1703/submission/165016289>  
→ [Reply](#)

Spinner4177

▲ 0 ▼

9 days ago, # | ☆ ← Rev. 2 ▲ 0 ▼  
anyone please tell me my mistake in problem G 165412847

[anyone please tell me my mistake in problem C](#)[→ Reply](#)

tushar\_kumar

9 days ago, <#> | [star](#)[▲ 0](#) [▼](#)

Call a pair good if it satisfies the condition. Let's split the inequality into three parts:  $a_i < i$ ,  $i < a_j$ ,  $a_j < j$ .



sheikhfoysal

Note that if  $a_i \geq i$  for any  $i$ , then it can't be an element of a good pair, because it fails the first and third conditions.

my question is: how does it fail the third condition?

[→ Reply](#)

divyansh\_sahu

7 days ago, <#> | [star](#)[▲ 0](#) [▼](#)

I intuitively got the solution for G but was unable to prove it. The proof in the editorial is great!

[→ Reply](#)5 days ago, <#> | [star](#)[← Rev. 4](#)[▲ 0](#) [▼](#)

<https://codeforces.com/contest/1703/submission/165888535>

I solved the question Double strings using a map but I still didn't get how the time complexity is  $O(l^*n\log n)$  According to my code time complexity would be

$\log n \rightarrow$  to insert elements in map

vk20july

$n \rightarrow$  outer loop  $l \rightarrow$  maximum length of input string  $l \rightarrow \text{substr}(\text{inbuiltfunction})$  Total  $TC \rightarrow \log n + n^*(l^*) \rightarrow (l^2)^*(n)$

[→ Reply](#)

[Codeforces](#) (c) Copyright 2010-2022 Mike Mirzayanov

The only programming contests Web 2.0 platform

Server time: Aug/01/2022 12:10:54 (i1).

Desktop version, switch to [mobile version](#).

[Privacy Policy](#)

Supported by

