CODEFORCES β
Sponsored by Telegram

SHawnHardy | Logout

HOME CONTESTS GYM PROBLEMSET GROUPS RATING API VK CUP CROC

ILYALOS BLOG TEAMS SUBMISSIONS GROUPS CONTESTS PROBLEMSETTING

IlyaLos's blog

Editorial for Codeforces Round #346 (Div. 2)

By IlyaLos, history, 6 days ago, translation,

659A — Round House

The answer for the problem is calculated with a formula $((a - 1 + b) \bmod n + n) \bmod n + 1$.

Such solution has complexity $O(1)$.

There is also a solution with iterations, modelling every of $|b|$'s Vasya's moves by one entrance one by one in desired direction, allowed to pass all the tests.

This solution's complexity is $O(|b|)$.

659B — Qualifying Contest

Let's consider the participants from every region separately. So for every region we just need to sort all of its participants by their score in non-increasing order. The answer for a region is inconsistent if and only if the score of the second and the third participant in this order are equal, otherwise the answer is the first and the second participant in this order.

The solution complexity is $O(n\log n)$.

659C — Tanya and Toys

Our task is to take largest amount of toys Tanya doesn't have yet the way the sum of their costs doesn't exceed m. To do that one can perform greedy algorithm: let's buy the cheepest toy Tanya doesn't have at every step, while the amount of money left are sufficient to do that. The boolean array $used$ can be a handle in that, storing $true$ values in indices equal to toy types which Tanya does have at the moment. As soon as 10^9 money is sufficient to buy no more than 10^5 toys ($\frac{10^5\times(10^5+1)}{2}\geq 10^9$), $used$ is enough to be sized 2×10^5 (we won't buy the toys with types numbered greater). So we just need to iterate over the number of type we want to buy, and if corresponding value in $used$ is equal to $false$, we should buy it, otherwise we can't.

The solution complexity is $O(n+\sqrt{m})$.

One can use the <> data structure (C++ \texttt{std::set}, for example), for storing the types Tanya has at the moment. In this case the complexity is $O(n+\sqrt{m}\log n)$.

659D — Bicycle Race

From the track description follows that Maria moves the way that the water always located to the right from her, so she could fall into the water only while turning left. To check if the turn is to the left, let's give every Maria's moves directions a number: moving to the north — 0, moving to the west — 1, to the south — 2 and to the east — 3. Then the turn is to the left if and only if the number of direction after performing a turn dir is equal to the number before performing a turn $oldDir$ plus one modulo 4 $(dir\equiv oldDir+1\pmod 4)$.

This solution has complexity $O(n)$.

→ Pay attention

Before contest
Educational Codeforces Round 11
3 days

Like 41 people like this.

→ SHawnHardy

Rating: 1555
Contribution: 0

- Settings
- Blog
- Teams
- Submissions
- Favourites
- Talks
- Contests

SHawnHardy

→ Top rated

#	User	Rating
1	tourist	4126
2	TooDifficult	3553
3	Petr	3424
4	rng_58	3125
5	izrak	3095
6	jcvb	3058
7	Um_nik	3007
8	vepifanov	2999
9	Egor	2991
10	RomaWhite	2985

Countries | Cities | Organizations View all →

→ Top contributors

#	User	Contrib.
1	Errichto	168
2	Zlobober	164
3	Petr	162
4	Edvard	157
5	Xellos	155
6	amd	150
7	chrome	146
8	Swistakk	145
9	rng_58	142
10	Rubanenko	140

View all →

→ Find user

One can solve this problem in alternative way. Let the answer be equal to x (that means that the number of inner corners of 270 degrees equals x, but the number of inner corners of 90 degrees to $n - x$). As soon as the sum of the inner corners' values of polygon of n vertices is equal to $180 \times (n - 2)$, then $x \times 270 + (n - x) \times 90$ equals to $180 \times (n - 2)$. This leads us to $x = \frac{n-4}{2}$, being the answer for the problem calculated in $O(1)$.

659E — New Reform

One should notice, that for every connected component of the graph the problem could be solved independently, so we just need to solve the problem for any connected graph.

Let this connected graph (of n vertices) contain $n - 1$ edge (such is called a tree). If one maintain a DFS from any of its vertex, every edge will be oriented, and each of them could given to its ending vertex, this way every vertex (except the one we launched DFS from, that is the root) will be satisfied by an edge. In this case the answer is equal to 1.

Let's then deal with a case when the graph contains more than $n - 1$ edges. This graph contains at least one cycle. Let's take arbitrary vertex from any of the cycles and launch a DFS (as above) from it. All vertices except chosen will be satisfied, so we are to give an edge to the chosen vertex. As soon as chosen vertex belongs to a cycle, at least one of its edge will not be taken to account in the DFS, so it can be given to a root. This way all the vertices will be satisfied.

Now we are able to solve the task for any connected graph, so we are to divide the graph into a connected components — this can be easily done by DFS or BFS.

The solution complexity is $O(n + m)$.

659F — Polycarp and Hay

In this task one should find a connected area, in which the product of the minimum value of the cells and the number of the cells is equal to k. To find such, let's sort all the $n \times m$ cells by their values by non-increasing order. Then we will consecutively add them in this order one by one, maintaining a connected components on their neighbouring relations graph. It's enough to use Disjoint Set Union structure to do so, storing additionally size of every component.

Let $a_{i,j}$ be the last added element in some component id, so $a_{i,j}$ has the minimum value among all the cells in component id (according to our ordering). If $a_{i,j}$ does not divide k, then the component id could not consist of desired area. Otherwise ($a_{i,j}$ is divisor of k), let's find $need = k / a_{i,j}$ — desired area size (if it contains $a_{i,j}$), and if CNT_{id} is not less than $need$, then the component id contains desired area, which could be easily found by launching a DFS in a neighbouring relation graph from $a_{i,j}$, visiting only the $a_{p,q} \geq a_{i,j}$ and exactly $need$ of them.

The solution complexity is $O(n^2 \log n)$.

659G — Fence Divercity

Let the answer for the problem be the sum

$$\sum_{l=1}^{n} \sum_{r=l}^{n} calc(l, r),$$

where $calc(l, r)$ is the number of ways to cut the top part of the fence the way its leftmost part is in position l and the rightmost in position r. If $l = r$, that is the case when the cutted top part consists of part of only one board, then $calc(l, r) = h_l - 1$.

Let $r > l$, then

$$calc(l, r) = \min(h_l - 1, h_{l+1} - 1) \times \min(h_{r-1} - 1, h_r - 1) \times \prod_{i=l+1}^{r-1} \min(h_{i-1} - 1, h_i - 1, h_{i+1} - 1).$$

In other words, the number of ways to cut the top part of some board is equal to minimum value among heights of it and its neighbours minus one, otherwise the cutted part will be

Handle:

Find

→ Recent actions

chaosagent → USACO US Open

Boxer → Unable to open topcoder contestApplet(Java version issue)

waterfalls → I mean... I'm not complaining (rating bug)

tweety → List of IOI 2016 Participants

FamIsProud → Comparator functions making my life difficult

Al.Cash → Efficient and easy segment trees

aropan → 6th BSUIR Open Programming Contest

ahmed_aly → Google Code Jam 2016 Warmups

MehrdadAP → Got stuck in an interesting combinatorial problem

maximaxi → Where can we find HackerEarth ratings?

kAc → Mo's Algorithm

Aksenov239 → Editorial for Round #118

Serh.kry → When java coders need multimap/multiset

King_Alexander → The Missing Differential

Nickolas → April Fools Day Contest 2016: editorial

sidchelseafan → Unable to view complete user submission history on SPOJ

raj454raj → StopStalk: Tool to maintain your algorithmic progress

IlyaLos → Editorial for Codeforces Round #346 (Div. 2)

suraj021 → BITMASKS — FOR BEGINNERS

NeverSayNever → Best Book for learning java

SkullSkin → High price for OpenCup's login

dtalamas24 → Which has been your favorite IOI?

AmSen → Iran IOI team 2016

xblwyc → Any data structure supporting log(n) update, delete and index query?

sahasumit288 → Floating To Integer problem:

Detailed →

> inconsistent. Leftmost board and rightmost board are considered separately, because each of them does have only one neighbouring board. So the answer looks like
>
> $$\sum_{i=1}^{i=n}(h_i-1)+\sum_{l=1}^{n}\sum_{r=l+1}^{n}\min(h_l-1,h_{l+1}-1)\times\min(h_{r-1}-1,h_r-1)\times\prod_{i=l+1}^{r-1}\min(h_{i-1}-1,h_i-1,h_{i+1}-1).$$
>
> The first summand is easy to calculate, so let's take a look at the second. Let us modify it as the following:
>
> $$\sum_{r=2}^{n}\min(h_{r-1}-1,h_r-1)\times\sum_{l=1}^{r-1}\min(h_l-1,h_{l+1}-1)\times\prod_{i=l+1}^{r-1}\min(h_{i-1}-1,h_i-1,h_{i+1}-1).$$
>
> Let
>
> $$S(r)=\sum_{l=1}^{r-1}\min(h_l-1,h_{l+1}-1)\times\prod_{i=l+1}^{r-1}\min(h_{i-1}-1,h_i-1,h_{i+1}-1).$$
>
> Let's take a look how does the $S(r)$ change after increasing r by one:
>
> $$S(r+1)=S(r)\times min(h_{r-1}-1,h_r-1,h_{r+1}-1)+min(h_r-1,h_{r+1}-1).$$
>
> This way this sum is easy to maintain if consecutively increase r from 2 to n.
>
> The solution complexity is $O(n)$.

+80 IlyaLos 6 days ago 71

Comments (71)

Write comment?

kingofnumbers

6 days ago, # | +60

Thanks for nice problems и fast Разбор

→ Reply

IlyaLos

6 days ago, # ^ | +12

Fixed. Thank you)

→ Reply

vsanjay_nitdgp

6 days ago, # | +1

Thanks for the lightning speed EDITORIAL.

→ Reply

h.hamza

6 days ago, # | 0

Nice round thank you :D

→ Reply

135678942570

6 days ago, # | +7

In F problem, I haven't seen the limit that must have an unchanged cell... So I wrote a code in O(sqrt(min(k, 1e15)).. When k > 1e15 print "NO", then enumerate the factor of k, use dsu to solve it, and got WA4. After contest I found the limit and add it then accepted...sad..

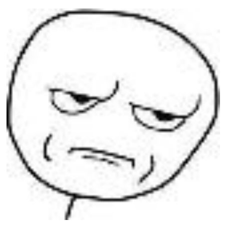
→ Reply

Flavius

6 days ago, # | +20

The best contest i've seen so far, I really liked the problemset and the fact that is was a 7 problem round instead of the regular one. Great job!

→ Reply



SohiebMohamed

6 days ago, # | ☆ 0

why this 17050130 gets WA in problem E?!!

any one can provide a small case which make it fail ?!

→ Reply

HustleHard

6 days ago, # ^ | ☆ 0

Problem should be solved for every component independently.

```
6 6
1 2
2 3
3 4
4 1
1 3
5 6
```

→ Reply

numbertheorist17

6 days ago, # ^ | ☆ +7

I believe this input is a small one where your program fails. There are two connected components: one is a tree and the other is a super well-connected component. The main idea for this test is that the degrees for some vertices are large enough so that when you subtract m edges, then you will not end up with any zeros.

```
10 13
1 2
1 3
1 4
1 5
1 6
2 3
3 4
4 5
5 6
6 2
7 8
8 9
9 10
```

→ Reply

HustleHard

6 days ago, # ^ | ☆ +26

Yea, you don't need to read other comments, better post the same idea that i've already posted but with two times bigger test, good job.

→ Reply

wrick

6 days ago, # | ☆ +3

I feel guilty for doing this 17058751 for 659D - Bicycle Race:

- I simply used `java.awt.Polygon` class to create the lake polygon.
- If Maria misses turn `i`, she would proceed by `dx = signum(x(i) - x(i+1))`, `dy = signum(y(i) - y(i + 1))`
- Then I simply checked if lake polygon contains `(x(i) + 0.5*dx, y(i) + 0.5*dy)`

→ Reply

Flavius

6 days ago, # ^ | ☆ +9

Well i guess that's still more ethical than "cout<<(n-4)/2";

→ Reply

dankmagician

6 days ago, # ^ | ☆ 0

Can anybody explain why that is correct?

→ Reply

Flavius

6 days ago, # ^ | ☆ 0

No idea

→ Reply

zydeoN

6 days ago, # ^ | ☆ +6

It's all explained in the editorial, but:
- assume that for all polygons (convex or concave) the sum of internal angles is $180(n-2)$
- for this problem, there are only internal angles of 270^o and 90^o and we need to count how many (x) of the former ones exist:

$$180(n-2) = 270x + 90(n-x)$$

→ Reply

GaoRed

6 days ago, # ^ | ☆ 0

I used ray casting with the same approach...

→ Reply

gandu007

6 days ago, # | ☆ ← Rev. 2 0

Hello everyone , Why testcase 53 is not correct in my answer since my answer totally matched with jury answer here is the link... for problem 2

→ Reply

afaxnraner

6 days ago, # ^ | ☆ 0

You have to check if `vec[i].size() > 2` before accessing the third element via `vec[i][2]` . Otherwise you may access unallocated memory. I made the same mistake...

→ Reply

thuustalu

6 days ago, # ^ | ☆ 0

Why does this only seem to fail on test 53? Even the first test case has a region with 2 people.

→ Reply

ToTheStars

6 days ago, # ^ | ☆ 0

Suppose size is 2. Then suppose that $vec[i][0] = vec[i][1]$. Then you will look for $vec[i][2]$ to compare with $vec[i][1]$, but that element might not exist (causing **Runtime Error**). In samples $vec[i][0] > vec[i][1]$, so you didn't need the third element.

→ Reply

alcinos

6 days ago, # | ☆ +6

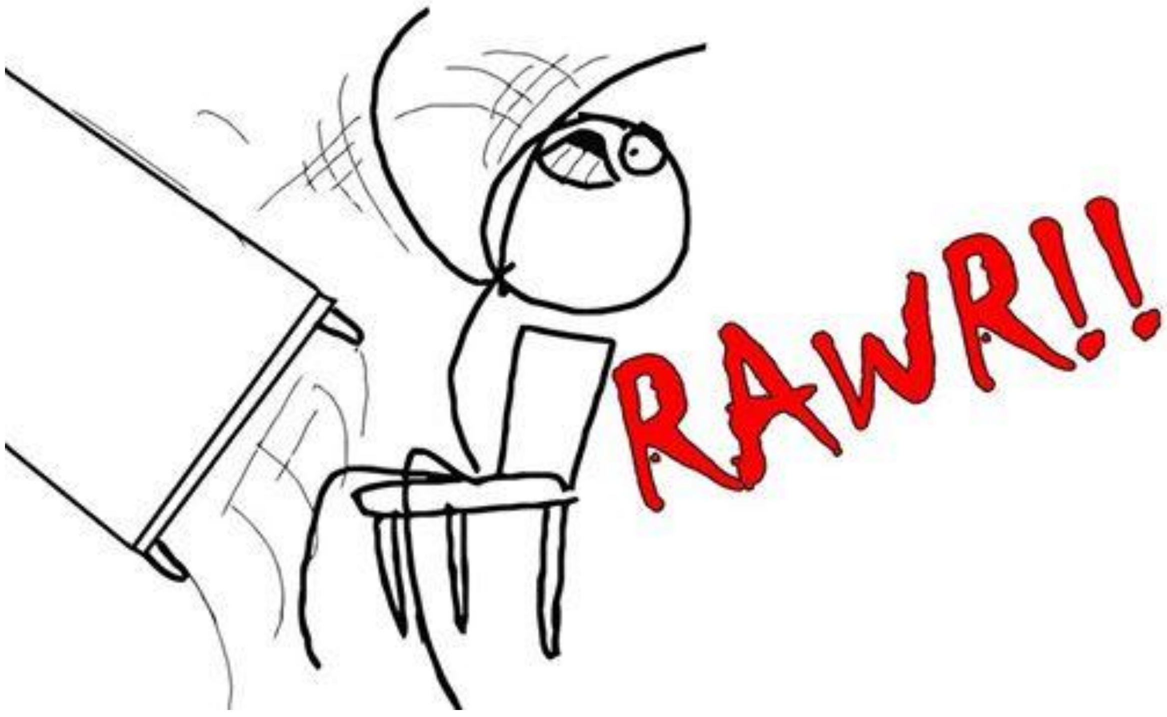
There is a parsing error in the editorial of F

→ Reply

6 days ago, # | ☆ 0

I solved D by finding point inside a polygon . I had to code almost from the scratch because somehow geeksforgeeks code was not working -_- . And after seeing this O(1) solution.....

dmehrab06



→ Reply

SarvagyaAgarwal

6 days ago, # ^ | ☆ ▲ 0 ▼

Happened with me too !
I found a working code though (around 7 lines)

```
int pnpoly(int nvert, float *vertx, float *verty,
float testx, float testy)
{
  int i, j, c = 0;
  for (i = 0, j = nvert-1; i < nvert; j = i++) {
    if ( ((verty[i]>testy) != (verty[j]>testy)) &&
         (testx < (vertx[j]-vertx[i]) * (testy-
verty[i]) / (verty[j]-verty[i]) + vertx[i]) )
       c = !c;
  }
  return c;
}
```

→ Reply

rajuknit

5 days ago, # ^ | ☆ ▲ 0 ▼

Hey,,please give me the link for this code's tutorial

→ Reply

SarvagyaAgarwal

5 days ago, # ^ | ☆ ▲ 0 ▼

Here You Go :D

→ Reply

5cm_per_second

6 days ago, # | ☆ ▲ 0 ▼

Can anybody tell me why i have a WA in test 15 of problem B?

→ Reply

Elise

6 days ago, # ^ | ☆ ▲ +4 ▼

The problem is where you print the answer.

When there are only 2 people in a region the **if** statement access an irregular position in memory.

→ Reply

ericvalls

6 days ago, # ^ | ☆ ▲ 0 ▼

I made this check and I got WA in test 15... Can someone tell me why, please?

→ Reply

5cm_per_second

6 days ago, # ^ | ☆ 0

oh, true! Thank you!

⟶ Reply

-Morass-

6 days ago, # ^ | ☆ 0

Hello,

sure, it is because you forgot to check size of vector

make the checking line:

if (v[i].size()<3u||v[i][1].p != v[i][2].p)

⟶ Reply

5cm_per_second

6 days ago, # ^ | ☆ +3

Thank you!

⟶ Reply

Mucosolvan

6 days ago, # | ☆ 0

Could anyone help me with that http://codeforces.com/contest/659/submission/17057892 ?

⟶ Reply

ielenik

6 days ago, # ^ | ☆ ← Rev. 2 0

BFS do not check for one cell solution.

⟶ Reply

uptimist

6 days ago, # | ☆ 0

My solution for problem D is to count the number of left turn and the number of right turn then take the minimal of them. I have no idea why it works. It was a desperate attempt. Can somebody provide a proof? Thank you!

My submission: http://codeforces.com/contest/659/submission/17055311

⟶ Reply

NYU

6 days ago, # ^ | ☆ ← Rev. 2 0

My guess is number_of_right_turns = number_of_left_turns + 3

So number_of_right_turns is always greater than number_of_left_turns.

You are actually taking number_of_left_turns.

Just cout << posi << endl; also get AC.

⟶ Reply

alexwice

6 days ago, # | ☆ 0

Can someone explain problem F in more detail please? Basically for each cell with value v, you want to know C(v) = how many cells are connected to it that are at least v. When you sort by decreasing value and DSU, won't your calculation of C(v) for cells with the same value be wrong? For example this grid:

2 4 5

6 4 3

⟶ Reply

NYU

6 days ago, # | ☆ ▲ 0 ▼

I didn't notice that in problem D we may only drop into water on left turn and solved the more generalized version of the problem.

The idea is we can record all the vertical and horizontal segments, and for each turning point we can count how many segments we may intersect if we keep going straight. If the count of segments is odd then this point is dangerous, and not while the count is even.

→ Reply

razibchamp_AIUB

6 days ago, # | ☆ ▲ 0 ▼

How does that formula come in problem number A ? Can somebody illustrate please ?

→ Reply

saquib2508

6 days ago, # ^ | ☆ ▲ 0 ▼

Forget the formula. Basically, all you need to do is to make a loop. When the system has 5 rooms, going 6 rooms front actually means going 1 room front, right? For backward movement, 2 rooms back means 3 rooms front. Just keep adding the number of total rooms until you get a non negative number. So taking mod is enough. Just to avoid 0'th position, as 5%5 or 6%6 makes zero, but you want to print 5 or 6, the mod operation is done in two steps. Eventually the -1 with a and the +1 at the end don't make any difference.

→ Reply

razibchamp_AIUB

5 days ago, # ^ | ☆ ▲ 0 ▼

Thank you for reply. Just keep adding the number of total rooms until you get a non negative number. I didn't understand this line.

→ Reply

MayankPratap

6 days ago, # | ☆ ▲ 0 ▼

In 2nd Problem, why is the output on terminal and Codeforces Judge different??

I got WA on 1st pretest again and again because of this..

My solution is :- http://codeforces.com/contest/659/submission/17065952

→ Reply

aeremin

6 days ago, # ^ | ☆ ← Rev. 2 ▲ 0 ▼

1) You are doing ios_base::sync_with_stdio(0);

2) Then read something C-style: scanf("%d %d",&n,&m);

3) Then read something C++-style: cin>>str>>reg>>score;

Due to 1), 2) and 3) both read same data. I.e. in 3 you read very first input line instead of second.

→ Reply

MayankPratap

5 days ago, # ^ | ☆ ▲ 0 ▼

Oh... I got it .. Thanks aeremin

→ Reply

6 days ago, # | ☆ ← Rev. 2 ▲ 0 ▼

Hello for problem D, I have a doubt what property does this test case violate:

findingLoveOfMyLife

```
5
1 1
1 2
-1 2
-1 1
1 1
```

→ Reply

ericvalls

6 days ago, # ^ | ☆ +3

The first vertex should be the one more in the south and in the west

→ Reply

findingLoveOfMyLife

6 days ago, # ^ | ☆ 0

Thanks!

→ Reply

bydefault

6 days ago, # | ☆ +3

First of all, thank you for the quick editorial!! I would like to mention 2 points regarding the editorial:

1. The phrase "as soon as" is used in places where "since" or simply "as" should be used. It is not that important(since the intention is very much clear), but feels odd! :P
2. For problem E, I did not understand the part "...as chosen vertex belongs to a cycle, at least one of its edge will not be taken to account in the DFS, so it can be given to a root. This way all the vertices will be satisfied." clearly. Can someone explain it in more details?

→ Reply

depevlad

6 days ago, # ^ | ☆ 0

It is perhaps slightly misworded. The key idea is that for any tree, we can pick any root and direct each of the edges so that only the chosen root of the tree is unaccessible. Therefore, if the program finds a cycle in a connected component, and we assign the root to any node of the cycle then one can perform the same algorithm, except this time there will be one extra edge incident to the picked root (otherwise, the cycle wouldn't exist). This extra edge can, of course, be directed to make the root accessible as well, thereby solving the problem.

→ Reply

bydefault

6 days ago, # ^ | ☆ ← Rev. 2 0

So, in short, does this mean the following?

1. The final answer is the sum of the answers for each individual connected component.
2. For a connected component, the answer is 1 if the component is a tree, 0 otherwise.

→ Reply

depevlad

6 days ago, # ^ | ☆ 0

Yes!

→ Reply

6 days ago, # | ☆ 0

Nice problems! Just a small observation: on problem B I got a little bit confused with the explanation of forming teams:

> *The team of each region is formed from two such members of the qualifying competition of the region, that none of them can be replaced by a schoolboy of the same region, not included in the team and who*

zydeoN

> *received a **greater** number of points*

Shouldn't that be **greater or equal**?

(Anyways, I figured it from the sample case where the first 3 have equal scores)

→ Reply

abdulrahman_bq

6 days ago, # | ☆ ▲ 0 ▼

can any one tell why am i getting RTE on problem B

this my submission

→ Reply

-Morass-

6 days ago, # ^ | ☆ ← Rev. 2 ▲ 0 ▼

Good day to you man!

The RTE is because of sort :'(you have to change comparator:

```
bool cmp(pair<string, int>s1, pair<string, int>
s2) {
    if (s1.second >= s2.second)
        return false;
    return true;
}
```

like this :)

+btw I'm convinced the part at the end is also wrong :/

I would change it like:

```
REP(i, 1, m + 1) {
        int size = mat[i].size();
        if (size==2||mat[i][size - 2].second !=
mat[i][size - 3].second) {
            size--;
            if (size - 2 >=0 ) {
                if (mat[i][size - 1].second != mat[i]
[size - 2].second) {
                    cout << mat[i][size ].first << "
" << mat[i][size - 1].first << endl;
                    continue;
                }
            } else {
                cout << mat[i][size ].first << " " <<
mat[i][size - 1].first << endl;
                continue;
            }
        }
        cout << "?" << endl;
    }
```

→ Reply

abdulrahman_bq

5 days ago, # ^ | ☆ ▲ 0 ▼

Thnx !! it worked !! but i can't understand why RTE ?!

→ Reply

5 days ago, # ^ | ☆ ← Rev. 2 ▲ 0 ▼

Well, sort comparator must be deterministic.

That means it must decide (in finite time) for each pair of elements, whether one of them is bigger (and which one) or whether they are equal.

-Morass-

a: A < B && !(B < A) means A < B

b: !(A < B) && B < A means A > B

c: !(A < B) && !(B < A) means A == B

anyway A<B && B<A (which could have been returned in your program) is "nothing" so the function could not determine, how to sort it :)

→ Reply

Ramtin

6 days ago, # | ☆ ▲ 0 ▼

Can someone explain solution for F in more details ?

→ Reply

renegade_warrior

6 days ago, # | ☆ ▲ 0 ▼

Problem G has a divide and conquer solution too ... :P *I wanst able to arrive at the formula stated in the editorial but divide and conquer was a bit more intuitive to me *

Link to submission- http://codeforces.com/contest/659/submission/17075860

→ Reply

attempter

5 days ago, # | ☆ ▲ 0 ▼

can anyone explain solution of problem E in detail.

→ Reply

marcospaulo

5 days ago, # ^ | ☆ ▲ +3 ▼

Every connected component of the graph, could be solved independently. Each connected component has at most 1 vertex with ending degree 0 after all (solving optimally) . If the connected component is a simple path or a tree root on a vertex v, we may notice that this component has 1 vertex with ending degree 0. If the connected component is cyclic, starting on any node we can organize the Edges in order to take no edges with ending degree 0. Imagine we have one conected component witch has a cycle and a path or a cycle and a tree or both :) , for example this: 1-2, 2-3, 3-1, 1-5, 5-6; Notice we can turn it into the problema of the cyclic connected componen. For this, we can also pick the vertex with in-out degree=1 and connect it. For example: 1-2 , 2-3 , 3-1, 1-5 5->6; 1-2,2-3,3-1, 1->5, 5->6; In this phase we have the same cycle! Therefore, we can run a dfs for each connect component and verify if it has any cycle. If it has add the answer by 1, if it has not, add the answer by 0.

→ Reply

iwamf

47 hours ago, # ^ | ☆ ▲ 0 ▼

If it has a cycle, we should not add 1 to the answer right? your last two lines confuse me

→ Reply

marcospaulo

44 hours ago, # ^ | ☆ ▲ 0 ▼

Sorry! You are right! if it has a cycle, we should NOT add 1 to the answer :)

→ Reply

vazzup

5 days ago, # | ☆ ▲ 0 ▼

Can someone fix the parsing error in editorial for F? Thanks in advance.

→ Reply

sanchayapol.lew

4 days ago, # | ☆ ▲ 0 ▼

Thanks. Some beautiful problems and elegant solutions out there. :)

→ Reply

Life_hacker

4 days ago, # | ☆ ▲ 0 ▼

There is O(1) solution for problem E. http://www.codeforces.com/contest/659/submission/17067296 Can anyone explain me this.

→ Reply

BasselBakr

4 days ago, # ^ | ☆ ▲ 0 ▼

It's not a real solution. It exploits the fact that many tests cases share the same solution and compares number of nodes, edges and first number in second line to identify the test case and print the solution accordingly.

You can easily hack it! ;)

→ Reply

Life_hacker

2 days ago, # ^ | ☆ ▲ 0 ▼

Means it has weak test cases.

→ Reply

BasselBakr

2 days ago, # ^ | ☆ ← Rev. 2 ▲ 0 ▼

Not necessarily, however, if the problem statement asked to print unreachable nodes it would be way more easier to solve it properly

→ Reply

Nams

2 days ago, # | ☆ ▲ 0 ▼

Please it would be really grateful if someone can explain editorial of Problem F.

→ Reply

PandaConda

40 hours ago, # ^ | ☆ ← Rev. 2 ▲ 0 ▼

You just need to launch dfs from all `i` and `j` that `k % a[i][j] = 0 and k / a[i][j] <= n * m` . In dfs count values that `v >= a[i][j]` . if value equals to k / a[i][j] you could find answer.

→ Reply

Nams

38 hours ago, # ^ | ☆ ▲ +5 ▼

Thanks I got your point but if we launch dfs from each i and j for the mentioned condition, won't the complexity be O(n*m*(n*m)) in worst case? How to make it efficient?

→ Reply

evandrix

17 hours ago, # | ☆ ▲ 0 ▼

Problem F has a missing text: `Otherwise ( Unable to parse markup [type=CF_TEX] )`

→ Reply