McDic's blog

Codeforces Round #589 (Div. 2) Editorial

By McDic, history, 18 hours ago,

Hello, I hope all of you enjoyed my contest!

1228A - Distinct Digits

Let's see how to check if all digits of x are different. Since there can be only 10 different numbers(0 to 9) in single digit, you can count the occurrences of 10 numbers by looking all digits of x. You can count all digits by using modulo 10 or changing whole number to string.

For example, if $x = 1217$, then occurrence of each number will be $[0, 2, 1, 0, 0, 0, 0, 1, 0, 0]$, because there are two 1s, single 2 and single 7 in x. So 1217 is invalid number.

Now do the same thing for all x where $l \le x \le r$. If you find any valid number then print it. Otherwise print -1.

Time complexity is $O((r - l) \log r)$.

[Behind story of A]

- There is a successful hack for A. I am really surprised.

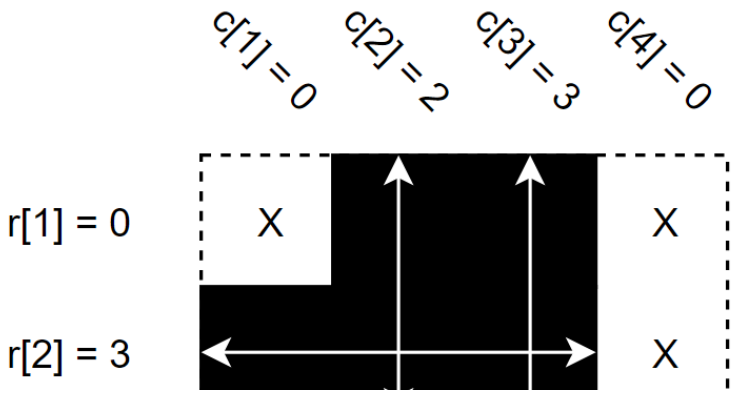
1228B - Filling the Grid

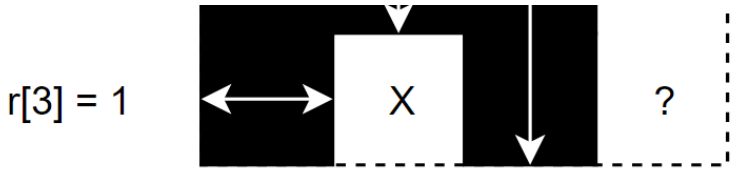
You can see some observations below;

- r and c values reserves some cells to be full, and some cells to be empty. Because they have to satisfy number of consecutive full cells in their row/column.
- If some cell is reserved to be full by some values and reserved to be empty by some other values, then it is impossible to fill grid. Let's call this kind of cell as invalid cell.
- If there is no invalid cell, then the answer is $2^{unreserved}$ where $unreserved$ means the number of unreserved cells, because setting state of unreserved cells doesn't affect validity of grid.

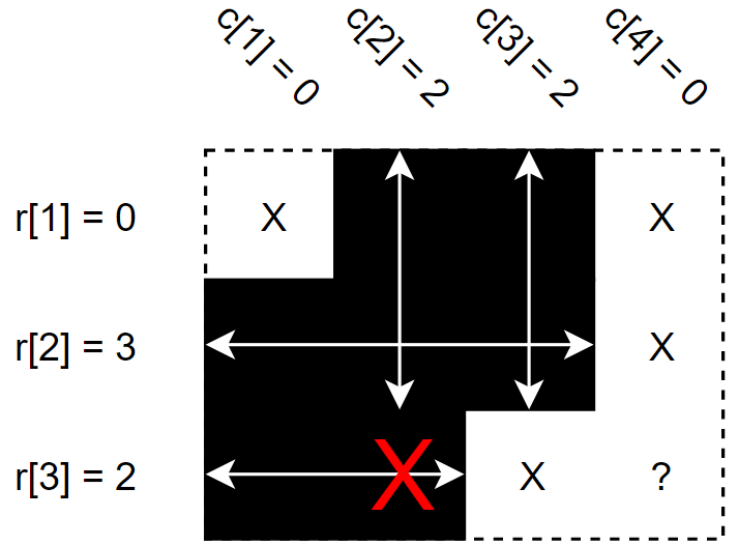
For easier understanding, please look at the pictures below.

- Black cells are reserved to be full by some r or c value.
- White *X* cells are reserved to be empty by some r or c value.
- White *?* cells are unreserved cells.
- Red *X* cells are invalid cells.





This is the explanation of the first example. There is 1 unreserved cell, so the answer is 2.



This is one of the impossible cases. That red *X* cell is reserved to be full by r_3, but reserved to be empty by c_2. So this is impossible.

Time complexity is $O(wh)$.

[Behind story of B]

- There is no behind story.

1228C - Primes and Multiplication

Let's say $h(x, p) = \log_p g(x, p)$, then $h(x, p) + h(y, p) = h(xy, p)$. Because if we describe $x = p^{h(x,p)}q_x$ and $y = p^{h(y,p)}q_y$, then $xy = p^{h(x,p)+h(y,p)}q_xq_y$.

Now let's go to the main step;

$$\begin{aligned}\prod_{i=1}^{n} f(x,i) &= \prod_{i=1}^{n}\prod_{p\in prime(x)} g(i,p)\\ &= \prod_{i=1}^{n}\prod_{p\in prime(x)} p^{h(i,p)}\\ &= \prod_{p\in prime(x)}\prod_{i=1}^{n} p^{h(i,p)}\\ &= \prod_{p\in prime(x)} p^{\sum_{i=1}^{n} h(i,p)}\\ &= \prod_{p\in prime(x)} p^{h(n!,p)}\end{aligned}$$

So we have to count $h(n!, p)$ for each p in $prime(x)$, and calculate exponents. You can count $h(n!, p)$ by following formula;

$$h(n!,p) = \sum_{k=1}^{\infty}\left\lfloor \frac{n}{p^k}\right\rfloor$$

Fortunately, since $h(n!, p)$ never exceeds n, we don't have to apply Euler's theorem here. You just have to be careful about overflow issue.

Roughly calculated time complexity is $O(\sqrt{x} + \log\log x \cdot \log n)$, because you use $O(\sqrt{x})$ to get prime divisors of x, and the number of distinct prime divisors of x is approximately $\log\log x$.

[Behind story of C]

- Initial version of C statement consists of tons of mathematical formula. CF team and testers requested me to reduce amount of mathematical formula.
- This problem was added before a week to the round. If there was no such C, the balance would be bad.
- Thanks for **dorijanlendvaj** , he improved test data for C a lot!

1228D - Complete Tripartite

You can make answer by following these steps;

1. If two vertices u_1 and u_2 are in same vertex set, there should be no edge between them. Otherwise, there should be edge between them.
2. If you choose any u as first vertex of specific vertex set, then you can simply add all vertices which are not directly connected to u in that vertex set.
3. Make 3 vertex sets by doing second step multiple times. If you can't make 3 sets or there is any vertex which is not in any vertex set, then answer is impossible.
4. If $m \neq |v_1| \cdot |v_2| + |v_2| \cdot |v_3| + |v_3| \cdot |v_1|$, then answer is impossible. $|v_i|$ means size of i-th vertex set.
5. For all vertices u_1 and u_2 from different vertex sets, if there is no direct connection between u_1 and u_2, then answer is impossible.
6. If you validated all steps, then current vertex set assignment is answer.

Make sure you are doing all steps. If you forget any of these steps, your solution will print wrong answer.

Time complexity is $O((n+m)\log n)$.

[Behind story of D]

- Same as C, I wrote tons of mathematical formula in D. After CF team's request, I reduced the amount of formula.
- This is my personal favorite problem among ABCDEF.

1228E - Another Filling the Grid

- $O(n^3)$ solution:
  Let $f(r, c)$ to be the number of filling grids of r rows, c incomplete columns, and $n - c$ complete columns. Incomplete columns means which doesn't contain 1 in already filled part, and complete columns means opposite. Now you can see that the formula can be described as below;
  - $f(r, 0) = (k^n - (k-1)^n)^r$ $(1 \leq r)$, because we don't have to care about minimum value of columns. However, there should be at least one cell which has 1.
  - $f(1, c) = k^{n-c}$ $(1 \leq c)$, because we have to fill 1 in all incomplete columns in that row. But, other cells are free.
  - $f(r, c) = (k^{n-c} - (k-1)^{n-c}) \cdot f(r-1, c) + k^{n-c} \cdot \sum_{c_0=1}^{c} \binom{c}{c_0} \cdot (k-1)^{c-c0} \cdot f(r-1, c-c_0)$ $(2 \leq r,\ 1 \leq c)$. Each part means number of cases when you select c_0 incomplete columns to be complete column in this row.

  The answer is $f(n, n)$.
- $O(n^2)$ and $O(n \log n)$ solution:
  Let $R[i]$ be the restriction of the i-th row having some value <= 1 and $C[i]$ the same but for columns. We want $\bigcap_{i=1}^{n} R[i] \cap C[i]$.

  Negate that expression twice, and we'll get $U - \bigcup_{i=1}^{n} \neg R[i] \cup \neg C[i]$. Using inclusion-exclusion this is:

  $$\sum_{i=0}^{n}\sum_{j=0}^{n}(-1)^{i+j} \cdot \binom{n}{j} \cdot \binom{n}{i} \cdot k^{n^2-n\cdot(i+j)+i\cdot j} \cdot (k-1)^{n\cdot(i+j)-i\cdot j}$$

  This is enough for $O(n^2 \log n)$ with fast exponentiation or $O(n^2)$ precomputing the

needed powers. To get $O(n \log n)$ note that we the second sum is a binomial expansion so the answer can be simplified to:

$$\sum_{i=0}^{n}(-1)^i \cdot \binom{n}{i} \cdot (k^{n-i} \cdot (k-1)^i - (k-1)^n)^n$$

[Behind story of E]

- **MikeMirzayanov** changed the name of problem before 10~20 minutes to the contest. It was "Minimum One" before.
- This problem was the easiest to prepare data. Just pure random and $k = 1$ is strong enough.
- I managed to create $O(n^3)$ pypy3 solution for E in 1 second lol.
- Thanks **tfg** for providing $O(n \log n)$ solution.

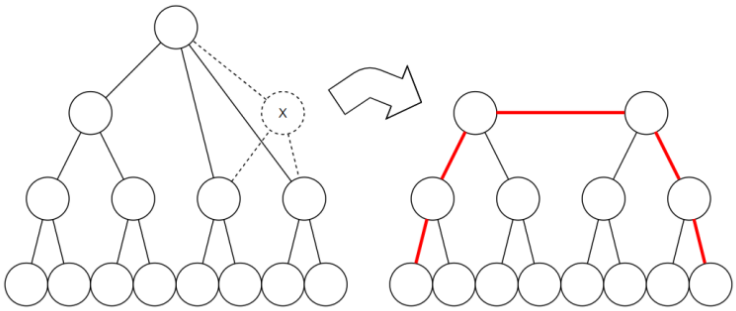
1228F - One Node is Gone

Let me suggest this observation;

- Root of generated tree should be one of middle of diameter. Because only 1 node is deleted from complete full binary tree.

So there are 3 valid cases;

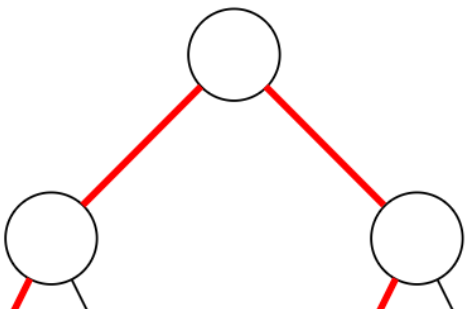
- The removed node is child of root. In this case, there are 2 answers(2 center nodes), diameter is decreased by 1 (odd), and tree looks like two complete full binary trees' roots are connected. You have to check if two center's subtrees are complete full binary tree.

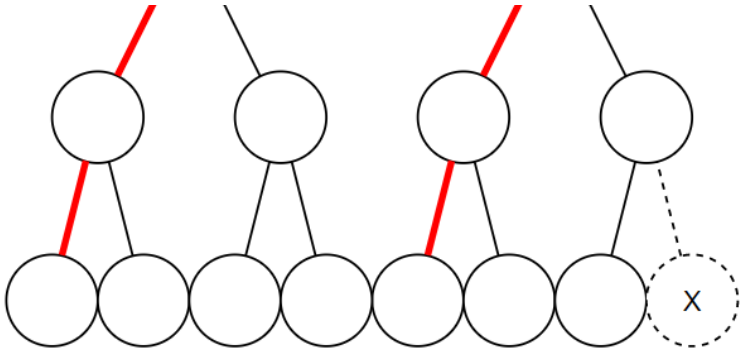


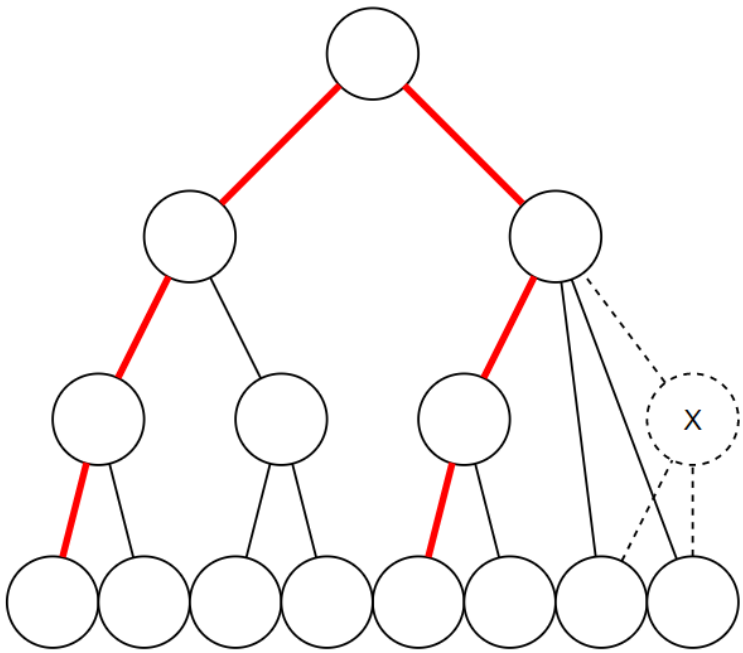
In this case, there are 2 answers, which are 2 centers of diameter.

- The removed node is leaf($n > 2$) or normal node. In this case, there is only 1 answer and 1 root node. Check if whole tree is complete full binary tree with 1 node error toleration. You can do case-handling by degree of nodes.
  - If non-root has degree 3, then this node is normal.
  - If non-root has degree 2 (error), then this node should be parent of removed leaf. You should check if this node's child node is leaf.
  - If non-root has degree 1, then this node should be leaf.
  - If non-root has degree 4 (error), then this node should be parent of removed normal node. This is the hardest case. I did this by checking depth of each child's subtree using DFS, then consider each tree to be complete and binary tree with no error, but with different depths.

  If you encountered multiple error nodes, then this tree is invalid. To check my exact approach, please look at my code.







In these cases, we can fix the center of whole tree by center of diameter.

To check if specific subtree is complete and full binary tree, you can use top-down recursive approach. Maybe you can use bottom-up approach by collapsing leaf nodes too, but it's very hard(at least I think) to check all conditions strictly.

Time complexity is $O(2^n)$. But you can solve this in like $O(n \cdot 2^n)$ or something bigger one.

[Behind story of F]

- This problem was the hardest to prepare data. We considered more than 10 types of trees to block various kind of WA solutions.
- I intended top-down error-toleration based case handling approach for this contest, but seems other approaches are also ok.
- Also thanks for **dorijanlendvaj** here, he is real MVP tester.

[Behind story of G (removed problem)]

- Nobody(including red testers) solved this problem for a week. This problem is saved as spare problem for another Div.1 contest.
- I love this beautiful problem than any other problems I ever made.

Thanks in advance!

Tutorial of Codeforces Round #589 (Div. 2)

mcdic

+263 McDic 18 hours ago 85

Comments (85)

Write comment?

shaf_wa_nur

17 hours ago, # | ☆ +73

The behind story to each problem (except B) is really interesting and new. Great round btw O_o

→ Reply

Aman.Gupta

17 hours ago, # | ☆ +125

Fast editorial, Fast system testing, Balanced Problemset, Nice Problems.
This is really one of the finest round of Codeforces.

→ Reply

AakashGoyal

16 hours ago, # ^ | ☆ 0

O_o

→ Reply

Bhatt21

17 hours ago, # | ☆ +3

D can be easily solved by hashing

→ Reply

sthfaceless

16 hours ago, # ^ | ☆ +4

can you explain it pls

→ Reply

Bhatt21

16 hours ago, # ^ | ☆ ← Rev. 3 +9

all the members of a particular set will have similar neigbours for ex in the first example 2,3 -> 1,4 5 6 , 4,5,6 -> 1,2,3 and 1 -> 2 3 4 5 6 so u can has these no.s([1,4,5,6],[1,2,3],[2,3,4,5,6]) and just check if u have total hash == 3

→ Reply

sthfaceless

16 hours ago, # ^ | ☆ 0

thx

→ Reply

S_Aditya

15 hours ago, # ^ | ☆ ← Rev. 2 0

in your code, for(ll i=1;i<=n;i++) pw[i] = 29*pw[i-1]; won't this part lead to overflow And can you please explain a bit your code

→ Reply

S_Aditya

14 hours ago, # ^ | ☆ 0

@Bhatt21

→ Reply

Thallium

12 hours ago, # ^ | ☆ +1

Here's my explanation: he uses hash so this step is to map all the points to numbers randomly, which makes it easier to judge if two points have the same set of neighbours. It doesn't matter if it overflows or not as long as all numbers are distrubuted randomly.

→ Reply

4 hours ago, # ^ | ☆ 0

Thanks

→ Reply

S_Aditya

12 hours ago, # ^ | ☆ 0

Alternatively a deterministic bitset solution, which I used during the contest.

→ Reply

tyrion__

16 hours ago, # | ☆ ← Rev. 2 +18

Very fast editorial, really appreciate it! But is there a proof for the solution of D?

→ Reply

woshinande

39 minutes ago, # ^ | ☆ ← Rev. 4 0

There isn't really a proof needed; it's just a constructive algorithm with lots of constraint checks. Step #4 is just a sanity check to make sure that step #5 won't take $O(n^2)$ time.

edit: Oh... you mean a proof that if the algorithm fails the answer is definitely impossible. That's a bit beyond my abilities at the moment

→ Reply

Spheniscine

16 hours ago, # | ☆ 0

61508959

I'm failing test case 13 in problem C. Can anyone tell me why? I'm unable to figure out

→ Reply

thedebator

16 hours ago, # ^ | ☆ 0

may be your find prime set algo don't recognize cases as 2 * 397

→ Reply

sthfaceless

16 hours ago, # | ☆ ← Rev. 2 +15

Problem G is the best problem I've ever seen !

→ Reply

fried-chicken

16 hours ago, # ^ | ☆ +20

how to solve G ? O_o

→ Reply

UNoobAble

16 hours ago, # ^ | ☆ -8

Considering the fact that the problem G is too complicated for most people, the solution below is only visble for the people who is clever enough .

My solution on problem G in O(n!):

→ Reply

fried-chicken

16 hours ago, # | ☆ +2

A really well-conducted contest. The system testing got over instantaneously and the editorial was posted without delay. We appreciate your effort.

I'm quite curious to wait for the ratings of the problem to see if C has a higher rating than B since I personally found B much harder than B for this contest.

During the contest, I thought D was a variation of the problem of finding a triangle in a graph. But, that problem is known to take at least $O(n^2)$ time. It's quite nice to see the solutions.

Here are all my solutions to this contest, in case anybody wants to refer them.

→ Reply

ghoshsai5000

15 hours ago, # ^ | ☆ ← Rev. 2 0

15 hours ago, # ^ | ☆ ← Rev. 2 0

We are not exactly multiplying the maximum power of p in n.

Firstly, we want to find the contribution of each *prime factor* in the product *independently* and *multiply* them.

What is the contribution of p ?

ghoshsai5000

p occurs in the product as many times as p has a multiple from $[1, n]$

p^2 occurs in the product as many times as p^2 has a multiple from $[1, n]$

And so on.

Basically, we are trying to find the number of 0 in $n!$ in base p

Please refer my explanation in GitHub and let me know if you have any doubts.

→ Reply

1600

15 hours ago, # ^ | ☆ 0

Thanks.

→ Reply

msdcoder7

13 hours ago, # ^ | ☆ 0

So, for finding the count from [1,n] for p, we have n/p — n/(p^2)...

isn't it?

→ Reply

ghoshsai5000

6 hours ago, # ^ | ☆ 0

We are not counting the number of multiples of p. We want the exponent of p.

Each multiple of p adds 1 to the exponent

Each multiple of p^2 adds 2 to the exponent

However, each multiple of p^2 is also a multiple of p. That is why, we only add n/p^2 once and not twice. Because it is already added while considering n/p

Similarly n/p^3 adds 3 to the exponent. But, we have already added it once in n/p and another time in n/p^2 so we just add n/p^3 once

→ Reply

shubham95

13 hours ago, # ^ | ☆ 0

in ur code in github u used power_mod() is it predefined b/c i didn't find its implementation on github

→ Reply

6 hours ago, # ^ | ☆ 0

akashpatil219

16 hours ago, # | ☆ 0

Can someone explain the intuition/proof for solution of D by the approach mentioned in the editorial or the hashing approach? Thanks in advance.

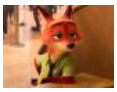
→ Reply

pleb

14 hours ago, # ^ | ☆ 0

every node from the same set has 2 properties, 1/ not connected to any node from same set, 2/ connected directly to all other nodes, so if we take adjacent list of each node from the same set it will be the same, so we hash the adjacent list after sort, same hash equals same set.

→ Reply



Andimeo

16 hours ago, # | ☆ 0

Is the definition of f(r, 0) for problem E incorrect? Or it's my problem with understanding English?

→ Reply

elManco

8 hours ago, # ^ | ☆ +1

It's correct actually.

$f(r, 0)$ are the ways to fill r rows and 0 incomplete columns (i.e. every column has already at least one 1).

Now, the idea behind the formula to calculate the ways to fill each row is: there are n squares in which in every one of them you can put every number from 1 to k (k^n). However, you are also counting ways in which there is no 1, which violates the condition of the problem. Therefore, just remove the number of ways which doesn't include any 1. It's the same idea, you have n squares and in each one of them you put every number from 2 to k ($(k-1)^n$).

So, the ways to fill each row is $k^n - (k-1)^n$ and to fill r rows is $(k^n - (k-1)^n)^r$.

→ Reply

siisback

50 minutes ago, # ^ | ☆ 0

Can you define once again, what is incomplete column?

→ Reply

elManco

44 minutes ago, # ^ | ☆ 0

An incomplete column is a column that so far it doesn't have any 1.

At the beggining you start with n incomplete columns as the grid is empty and in each step (filling one row) you can decrease the number of incomplete columns by placing $1's$ on them or you can keep it the same.

→ Reply

PvDzxc

16 hours ago, # | ☆ +6

D can be solved with some vector sorting (sort and sort). what a very strong vector! :D

Of course, Thanks McDic for the best CF round I have ever participate in!

→ Reply

spookywooky

16 hours ago, # | ☆ +3

I usually do not like a problem if I do not understand the tutorial, like C and E.

B is a problem where one can miss several things, resulting in late errors. With such problems it is nice to have strong pre-tests. Unfortunatly we do not have them here.

→ Reply

tfg

16 hours ago, # ^ | ☆ +17

It's almost impossible to make 0 solutions fail on systest, you're being biased and not seeing the big picture here, there were really few FST.

→ Reply

spookywooky

15 hours ago, # ^ | ☆ -19

Of course I am biased, because my B failed on test 21, which is annoying.

There is that 1x1 example (the second one). That could have been at least a 2x2 example without spoiling the problem in any way, but having a meaningful "negative" example.

However, if you are not willing to give such example, then at least make it a pre-test. I do not think that was because of "we cant test everything", I think it was intentionally.

→ Reply

ValeriyG

16 hours ago, # | ☆ ← Rev. 4 0

How actually the grid looks for this test of B?

```
4 4
2 0 3 1
1 3 0 3
```

I can't understand at all

After applying r_i's I've got

```
1 1 0 x
0 x x x
1 1 1 0
1 0 x x
```

I can't understand how to apply c4 because according to the problem statement in third row I have 1 1 1 0

```
1 1 0 1
0 1 x 1
1 1 1 ?
1 0 x 0
```

And why right answer is 0 if we have 2 unreserved cells, its M[2][3] and M[4][3]?

→ Reply

spookywooky

15 hours ago, # ^ | ☆ 0

The row descriptions says cell 4,3 must be white, the col description says it must be black. So, there is no solution, hence the answer is 0.

→ Reply

ValeriyG

15 hours ago, # ^ | ☆ 0

Thanks. I thought there is always a correct grid.

→ Reply

3 hours ago, # ^ | ☆ 0

rupav

15 hours ago, # | ☆ +8

Can someone please explain problem E tutorial — can't understand what's f(r, c) is here. Thanks.

→ Reply

8 hours ago, # ^ | ☆ +15

I explained a few comments above $f(r, 0)$. I think it would be nice to read it if it isn't completely clear because I will refer to that.

The first term, which I think it's really $(k-1)^c \cdot (k^{n-c} - (k-1)^{n-c}) \cdot f(r-1, c)$ means:

You have c incomplete columns and in this step, you won't decrease this number. It means that in the c incomplete columns you can put every number from 2 to k ($(k-1)^c$), the 1 is forbidden because this way you would decrease the number of incomplete columns.

Now, you can fill the remaning $n-c$ squares as you want as long as there is at least one 1 in the row, that is $k^{n-c} - (k-1)^{n-c}$ (I explained this in my previous comment) and now you have 1 row less but still c incomplete columns $f(r-1, c)$.

elManco

The second term

$$\sum_{c_o=1}^{c} k^{n-c} \cdot \binom{c}{c_o} \cdot (k-1)^{c-c_o} \cdot f(r-1, c-c_o)$$

is when you are reducing the number of incomplete columns by c_o which goes from 1 to c.

As you will place at least one 1 in this row to reduce the number of incomplete columns, the $n-c$ already completed columns are free to choose every number from 1 to k (k^{n-c}). Now, you can place the c_o ones in the c incomplete columns in $\binom{c}{c_o}$ ways. Among the $c - c_o$incompleted columns which will remain incomplete, you can choose every number from 2 to k only ($(k-1)^{c-c_o}$). And finally, you have 1 row less and $c - c_o$ incomplete columns $f(r-1, c-c_o)$.

Notice that the first factor is constant in the summatory so it can be placed outside.

→ Reply

namansinghal198

15 hours ago, # | ☆ +3

D: Check if the complement of the Graph (connect nodes i,j if they are not connected in the original graph) has exactly 3 completely connected components. This is the only check needed acc to me, sadly couldn't do it efficiently in time.

→ Reply

yonkoamit

12 hours ago, # ^ | ☆ 0

Can you explain why does it works and how did you get the intuition to this approach?

→ Reply

rupav

5 hours ago, # ^ | ☆ 0

Great approach. But making a complement graph, by checking for each pair, i&j, will give you TLE — O(n^2) approach.

→ Reply

4 hours ago, # ^ | ☆ ← Rev. 3 0

nytemusik

15 hours ago, # | ☆ 0

Where is the solution code for the One Node is Gone problem?

→ Reply

McDic

7 hours ago, # ^ | ☆ ← Rev. 2 0

I am sorry. I will add it later.

→ Reply

harshhx17

15 hours ago, # | ☆ +11

In the solution of E

```
Now you can see that the formula can be described as below;
```

I think in the third formula the first term should be multiplied with (k-1) ^ c

https://codeforces.com/contest/1228/submission/61516757

→ Reply

pk842

14 hours ago, # ^ | ☆ 0

I didn't get you. What I understand is we are filling all the 'c' incomplete columns here and so all of them have '1' and in remaining we can choose anything but there must be atleast one '1' in this row and so we have the first term as written there. So, where from that $(k-1)^c$ comes?

And also I have a doubt. If I understood the approach correctly, so whenever $c > 0$ we are sure that there are atleast 1 column which is incomplete and we are going to fill it here. So, isn't the first term should be K^(n-c) only?

→ Reply

harshhx17

11 hours ago, # ^ | ☆ 0

The first term is for the case when we keep the same number of incomplete columns, so we can place anything but 1 in those c columns (k-1)^c and among the ones that are complete, we must have atleast one 1. (k^ (n-c) — (k-1)^(n-c)) and then make a recursive call to (r-1,c)

McDic, Can you please verify this!

→ Reply

McDic

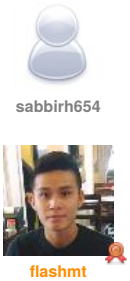
10 hours ago, # ^ | ☆ ← Rev. 2 0

▶ My code

→ Reply

110 minutes ago, # ^ | ☆ 0

```
f[r][c] = clean(kpower[n-c] - k1power[n-c]) * k1power[c] %
```



sabbirh654

15 hours ago, # | ☆ 0

didn't understood explanation of B.please help

→ Reply

flashmt

14 hours ago, # | ☆ +11

The formula in the $O(n^2)$ solution for E should be -1^{i+j}?

→ Reply

tfg

14 hours ago, # ^ | ☆ 0

Yes, sorry.

→ Reply

McDic

10 hours ago, # ^ | ☆ 0

I fixed. It will be reflected soon.

→ Reply

Smile.Forever

14 hours ago, # | ☆ 0

In 1228B - Filling the Grid, I'm trying to count all combinations (nCr) of unreserved cell. I'm not sure why combination is not the right way?? and why 2^unreserved_cell should be the output??

→ Reply

spookywooky

12 hours ago, # ^ | ☆ 0

You can make any of these cells black or white. So two possibilities per cell. Each one cell independent of all other cells. So if you got one more cell, the number of possibilities doubles.

→ Reply

Smile.Forever

3 hours ago, # ^ | ☆ 0

Thanks for clarification.

→ Reply

Smile.Forever

3 hours ago, # ^ | ☆ 0

I just forget it, total combination is = 2^n, What's a silly mistake I have done :|

→ Reply

Nson

13 hours ago, # | ☆ +3

Why step 4 is necessary for D? I thought it was sufficient to put vertices without a direct edge in the same set and checking if there is an edge between two vertices in the same set.

→ Reply

flashmt

13 hours ago, # ^ | ☆ 0

I didn't do that check in my solution but from what I guess, it's meant to check that there should be no edges between nodes in the same set. (Note that you can't iterate through every pair because the number of pair can be very large)

→ Reply

13 hours ago, ☆ | ☆ -12

joker70

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

const int MAX = 250 + 5;
const int MOD = 1e9 + 7;

#define dbg(a) cout << "-> " << #a << " = " << a << endl

int add (int a, int b) {
    return (a + b < MOD) ? (a + b) : (a + b - MOD);
}

int sub (int a, int b) {
    return (a >= b) ? (a - b) : (a - b + MOD);
}

int mul (int a, int b) {
    return (a * 1LL * b) % MOD;
}

int expo (int a, int b) {
    int ret = 1;
    while (b != 0) {
        if (b & 1) {
            ret = mul(ret, a);
        }
        a = mul(a, a);
        b >>= 1;
    }
    return ret;
}

int main() {
        vector<vector<int>> C(MAX, vector<int> (MAX));
    for (int n = 0; n < MAX; n++) {
        for (int r = 0; r <= n; r++) {
            if (n == r or r == 0) {
                C[n][r] = 1;
            }
            else {
                C[n][r] = add(C[n - 1][r], C[n - 1][r - 1]);
            }
        }
    }

    int n, k;
    scanf("%d %d", &n, &k);

    vector<int> x(MAX), y(MAX);
    x[0] = y[0] = 1;
    for (int i = 1; i < MAX; i++) {
        x[i] = mul(k, x[i - 1]);
        y[i] = mul(k - 1, y[i - 1]);
    }

    vector<vector<int>> dp(n + 1, vector<int> (n + 1));
    int val = sub(x[n], y[n]);
    for (int r = 1; r <= n; r++) {
        dp[r][0] = expo(val, r);
    }
    for (int c = 1; c <= n; c++) {
        dp[1][c] = x[n - c];
    }
    for (int r = 2; r <= n; r++) {
        for (int c = 1; c <= n; c++) {
```

```
                int ret = mul(dp[r - 1][c], sub(x[n - c], y[n - c]));
                for (int c0 = 1; c0 <= c; c0++) {
                    int now = mul(C[c][c0], y[c - c0]);
                    now = mul(now, dp[r - 1][c - c0]);
                    ret = add(ret, mul(now, x[n - c]));
                }
                dp[r][c] = ret;
            }
        }
        printf("%d\n", dp[n][n]);
        return 0;
}
```

Why this code gives me the incorrect output for test 2 in problem E? I just implemented the function the tutorial told me to. Can someone please help?
→ Reply

techgig0

11 hours ago, # ^ | ☆ +1

Man you are very active on cf(your graph says it) and you don't know how to use SPOILER or LINK in comments for your code. It is really annoying to see such huge codes in comments.
→ Reply

joker70

11 hours ago, # ^ | ☆ 0

Sorry. but I don't know how to do that.
→ Reply

sultanov_

5 hours ago, # ^ | ☆ 0

Just go to https://paste.ofcode.org or https://ideone.com (usually I use paste.ofcode.org 'cause it more easy to use) and paste your code, then you can just send to someone the link.

Like this -> https://paste.ofcode.org/329jEhW7par3TKGBpYxpsNK or just link the word -> **click**.

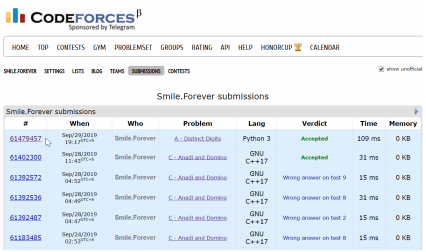
Hope it helps u.
→ Reply

Smile.Forever

2 hours ago, # ^ | ☆ +5

How to share your submission:

Example: Check out my submission 61479457:

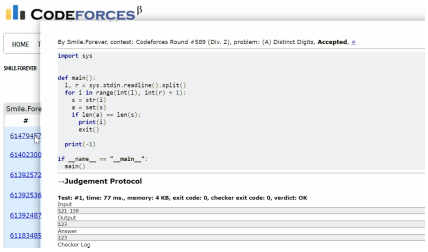


→ Reply

Smile.Forever

114 minutes ago, # ^ | ☆ ← Rev. 2 +5

You can share your code here:
CODE



→ Reply

techgig0

11 hours ago, # | ← Rev. 2 0

In problem D I missed 1 line and related it to 3-coloring problem. How stupid of me!

→ Reply

j2v

10 hours ago, # | ← Rev. 2 0

Can anyone explain what a "restriction" (in E tutorial) is? If we can intersect these "restrictions", then I guess it's a set. What are elements of these sets?

→ Reply

j2v

9 hours ago, # ^ | ← Rev. 3 +18

Ok, I think that I got it. I understand it in the following way. We are considering only grids of elements from $[1, k]$. R_i is the set of all grids where all elements in the i-th row are greater than 1 and C_j is the set of all grids, where all elements in the j-th column are greater than 1. So, $R_1 \cup R_2 \cup \ldots \cup R_n \cup C_1 \cup C_2 \cup \ldots \cup C_n$ is set of all bad grids (grids that have at least one row or column without 1), let's denote it as B. Let U be the set of all grids. In this problem we need to find the number of grids that have at least one 1 in each row and at least one 1 in each column. Therefore, we just take the set of all grids and subtract the set of all bad grids: $U \setminus B$.

Am I right? (I know that it differs a bit from what is written above. However, it still seems to be something equivalent)

→ Reply

tfg

8 hours ago, # ^ | 0

Correct!

→ Reply

rsFalse

10 hours ago, # | +6

In **A**, simply check which numbers do not have same digit pair. In Perl: `print +( grep !/(.).*\1/, -1, $l .. $r )[ -1 ]`

→ Reply

not_found99

6 hours ago, # | 0

Can anyone please explain problem B with code. Thanks.

→ Reply

rupav

5 hours ago, # ^ | ← Rev. 2 0

Just fix the matrix, according the constraints, take the matrix, as 3 state value —

- -1: not visited
- 0: visited, and must be white/empty
- 1: visited, and must be black/filled.

Now apply the constraints on all rows one by one. Then for each column, check if its possible to apply the constraints given for those columns. Once applied, the remaining -1 ' count, (say cnt) each can be filled with

either 1 or 0. Thus 2^cnt will be the answer.

Refer my submission: https://codeforces.com/contest/1228/submission/61514443

→ Reply

23 minutes ago, # ^ | ☆ 0

Possible solution of **B**.(Perl example)

Here:
X: not visited
0: visited, and must be white/empty
1: visited, and must be black/filled

Make an array of strings containing 'X'-es. `@strings = ( 'X' x w ) x h;`

rsFalse

Write a subroutine (function) which searches and replaces (`s///`) from beginning of a string.
SEARCH for exact number (depending on h_i or w_i) of consecutive 'X' or '1', and '1' must not follow (`(?!1)` as negative look-ahead), but any other symbol may follow (`(.)?` , saved as capture `$1`): `^[X1]{$fill->[$i]}(?!1)(.)?` .
REPLACE this with consecutive '1' followed by '0' (if `$1` defined): `1 x $fill->[$i] . ( defined $1 ? 0 : '' )` .

1. Apply subroutine for an array of strings, using values from array `h` .
2. Transpose strings.
3. Apply subroutine for an array of strings, using values from array `w` .

Count 'X'-es, and answer is 2^X with mod. If matching fails at any point, that means it is impossible to fill correctly, answer zero.

→ Reply

hash_mapper

6 hours ago, # | ☆ 0

In problem C, what i am doing is calculating the maximum number of divisible elements from **1 to n** for each power of **factor** where factor is the prime factor of **x**. However it is resulting in WA, here's my submission 61509357

→ Reply

sultanov_

5 hours ago, # | ☆ +5

Very nice round, problems was really good to solve and there was no bugs!

One of the interesting rounds I've ever solved.

Thank you McDic ^3^

→ Reply

timing

5 hours ago, # | ☆ ← Rev. 2 0

I'm sorry that the Tutorial for problem E in $O(n^3)$ must be wrong. How can the answer be $f(n)(n)$ in your define? I'm look forward to reading the correct Tutorial .

→ Reply

3 hours ago, # | ☆ 0

I guess, it'd be better if you had rather written $R[i]$ to be the set of matrices which have at least one 1 in its i-th row. It took me couple of minutes to realize what you actually meant.

memento

Also, I think the difficulty of problems didn't increase as it is supposed to. Maybe nlgn version of E could be moved to F and F to E. But, having both D and F in a single round is kind of not-so-interesting for the contestants. Like, you can figure out what the solution is at first sight but have to spend some boring time figuring out every bits and pieces and also the Δ difficulty of D and F is actually not that mush :(.

→ Reply

chowdhuryan1999

2 hours ago, # | ☆ 0

problem D https://codeforces.com/contest/1228/submission/61536068 i am geting wrong on testcase 6...help me to know which cases i am missing..

→ Reply

kimoyami

68 minutes ago, # ^ | ☆ ← Rev. 3 0

maybe you forget that three sets must not be empty? The reason I am wrong answer on testcase 6 is that.

→ Reply

chowdhuryan1999

64 minutes ago, # ^ | ☆ 0

No..that was checked..please go through my code once!

→ Reply