CODEFORCES
Sponsored by TON

|

shine_rocks | Logout

You have +353! Wow!

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP ICPC CHALLENGE

VLADOSIYA BLOG TEAMS SUBMISSIONS GROUPS CONTESTS PROBLEMSETTING

Vladosiya's blog

Codeforces Round 903 (Div. 3) Editorial

By **Vladosiya**, history, 3 weeks ago, translation,

1881A - Don't Try to Count

Idea: **Vladosiya**

Tutorial

Solution

1881B - Three Threadlets

Idea: **Gornak40**

Tutorial

Solution

1881C - Perfect Square

Idea: **myav**, **MikeMirzayanov**, **Vladosiya**

Tutorial

Solution

1881D - Divide and Equalize

Idea: **myav**

Tutorial

Solution

1881E - Block Sequence

Idea: **MikeMirzayanov**

Tutorial

1881E - Block Sequence

Let's use the dynamic programming approach: let $dp[i]$ be the number of operations required to make the segment from i to n beautiful. There are two possible ways to achieve this:

- Remove the element at position i and make the segment from $i+1$ to n beautiful, then $dp[i] = dp[i+1]+1$;
- Make the segment from $i + a_i + 1$ to n beautiful, then $dp[i] = dp[i + a_i + 1]$ (cases where $i + a_i + 1 > n$ need to be handled separately).

Solution

```
def solve():
    n = int(input())
    a = [int(x) for x in input().split()]
    dp = [n + 1] * n

    def get(pos):
        if pos > n:
            return n + 1
        if pos == n:
            return 0
        return dp[pos]
```

→ Pay attention

Before contest
Educational Codeforces Round 157 (Rated for Div. 2)
2 days
Register now »

→ shine_rocks

Rating: **890**
Contribution: **+6**

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

shine_rocks

→ Top rated

#	User	Rating
1	Benq	3833
2	jiangly	3720
3	Radewoosh	3714
4	orzdevinwang	3697
5	cnnfls_csy	3597
6	tourist	3557
7	-0.5	3545
8	inaFSTream	3478
9	fantasy	3468
10	ecnerwala	3466

Countries | Cities | Organizations View all →

→ Top contributors

#	User	Contrib.
1	adamant	176
2	awoo	165
2	BledDest	165
4	SecondThread	164
5	maroonrk	162
5	Um_nik	162
7	nor	158
8	-is-this-fft-	151
9	Geothermal	145
9	kostka	145

View all →

→ Find user

Handle:

Find

```
    dp[-1] = 1
    for i in range(n - 2, -1, -1):
        dp[i] = min(dp[i + 1] + 1, get(i + a[i] + 1))
    print(dp[0])


for _ in range(int(input())):
    solve()
```

1881F - Minimum Maximum Distance

Idea: DmitriyOwlet

Tutorial

1881F - Minimum Maximum Distance

Let's run a breadth-first traversal from any labeled vertex v_1 and find the farthest other labeled vertex v_2 from it. Then we run a traversal from v_2 and find the farthest labeled vertex v_3 from it (it may coincide with v_1). Then the answer is $\lceil \frac{d}{2} \rceil$, where d is the distance between v_2 and v_3. It is worth considering separately the case when there is only one labeled vertex in the tree. Then the answer is 0.

Solution

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

using namespace std;

int n;
vector<vector<int>> g;

void dfs(int v, int p, vector<int> &d){
        if(p != -1) d[v] = d[p] + 1;
        for(int u: g[v]){
                if(u != p){
                        dfs(u, v, d);
                }
        }
}

int main(){
    int t;
        cin>>t;
        while(t--){
            int k;
                cin>>n>>k;
                g.assign(n, vector<int>(0));
                vector<int> marked(k);
                for(int &e: marked) cin >> e, --e;
                for(int i=1;i<n;i++){
                        int u, v;
                        cin >> u >> v;
                        --u, --v;
                        g[u].push_back(v);
                        g[v].push_back(u);
                }
                if(k==1){
                        cout<<0<<"\n";
                        continue;
                }
                vector<int> d1(n);
                dfs(marked[0], -1, d1);
                int mx = marked[0];
                for(int e: marked){
                    if(d1[e] > d1[mx]) mx = e;
                }
                vector<int> d2(n);
                dfs(mx, -1, d2);
```

→ Recent actions

drugkeeper → Comprehensive guide to avoid TLE in python

coderdhanraj → Finally Candidate Master!

Yandex → Yandex Cup 2023

74TrAkToR → Codeforces Round #904 (Div. 2) Editorial

127.0.0.1 → Codeforces Round 907 (Div. 2) Editorial

jli505 → IOI Gold Medalist Speaker Event

noomaK → IEEEXtreme 17.0 Problems Discussion

rivalq → Codeforces Round #793 (Div. 2) Editorial

Cristofor → JBOI 2023 Teams

Jon.Snow → CSES Tree section editorial

BledDest → Educational Codeforces Round 147 — Editorial

LucaLucaM → BOI (balkan) 2023 mirror

Loser_ → CSES Sorting and Searching section editorials

_Firdavs_Aminov_ → Dark theme Codeforces

shn271 → Common Preliminary Round for ICPC India Regionals?

Ormlis → Codeforces Round #829 Editorial

127.0.0.1 → Codeforces Round 907 (Div. 2)

handleisinuse → CF Extensions Needed

MikeMirzayanov → Please, read this

hetanhnandre → Educational Geometry Problems

ArSarapkin → Codeforces rating graphic for some users

Imakf → Codeforces Round 906 (Div. 1, Div. 2)

m_devansh → Help Needed ! The CP Snippets Treasure Hunt

74TrAkToR → Codeforces Round #905 (Div. 1, Div. 2, Div. 3) Editorial

purp4ever → Teams Qualified for Africa and Arab Collegiate Programming Championship — ACPC 2023 (+teens)

Detailed →

```
                mx = marked[0];
                for(int e: marked){
                    if(d2[e] > d2[mx]) mx = e;
                }
                cout << (d2[mx] + 1) / 2 << "\n";
        }
        return 0;
}
```

1881G - Anya and the Mysterious String

Idea: **Gornak40**

Tutorial

1881G - Anya and the Mysterious String

Let's make two obvious observations about palindromes of length at least 2:

- Palindromes of length $2n$ contain a palindrome substring $[n \ldots n+1]$;
- Palindromes of length $2n+1$ contain a palindrome substring $[n \ldots n+2]$.

Now we need to learn how to track only palindromes of length 2 and 3. Let's call an index bad if a palindrome of length 2 starts from it, and terrible if a palindrome of length 3 starts from it. We will store the bad and terrible indices in a pair of `std::set`.

Let's assume that there are no modification queries and we need to check the substring $[l \ldots r]$ for beauty. If there exists a bad index i $(l \le i \le r-1)$ or a terrible index j $(l \le j \le r-2)$, then the substring is not beautiful; otherwise, it is beautiful. This can be checked using binary search on the set.

Now let's learn how to make modifications. Notice that palindromes do not appear or disappear inside a segment, but they can appear or disappear at its boundaries. Let's use a data structure that can add values on a segment and retrieve a value at a point. If such a structure exists, we will add $x \mod 26$ to the segment $[l \ldots r]$, and then process the nearest 10 indices to l and the nearest 10 indices to r. We can describe more precisely which bad and terrible indices should be processed, but ~~the author is lazy~~ it does not affect the solution, because the number of such indices is still $O(1)$. When processing the indices, we use point queries and insertion/deletion operations in `std::set`.

Now we need to implement such a data structure. The author suggests using a Fenwick tree on a difference array. A segment tree with lazy propagation will also work. This gives a solution with $O((n+q)\log n)$ time complexity.

Solution

```
#include <iostream>
#include <string>
#include <set>
#include <cstring>
#define int long long

using namespace std;

const int L = 26;
const int MAXN = 200200;

int n, m;
string s;
set<int> M2, M3;
int fen[MAXN];

void fenadd(int i, int x) {
        x = (x % L + L) % L;
        for (; i < n; i |= (i + 1))
                fen[i] = (fen[i] + x) % L;
}

int fenget(int i) {
        int ans = 0;
        for (; i >= 0; i = (i & (i + 1)) - 1)
                ans = (ans + fen[i]) % L;
        return ans;
}
```

```
void relax(int l, int r) {
        l = max(l, 0LL);
        r = min(r, n);
        for (int i = l; i + 1 < r; ++i) {
                int c1 = fenget(i);
                int c2 = fenget(i + 1);

                if (c1 == c2) M2.insert(i);
                else M2.erase(i);

                if (i + 2 >= r) continue;

                int c3 = fenget(i + 2);
                if (c1 == c3) M3.insert(i);
                else M3.erase(i);
        }
}

void build() {
        M2.clear();
        M3.clear();
        memset(fen, 0, n * sizeof(int));
        fenadd(0, s[0] - 'a');
        for (int i = 1; i < n; ++i) {
                fenadd(i, s[i] - s[i - 1] + L);
        }
        for (int i = 0; i + 1 < n; ++i) {
                if (s[i] == s[i + 1]) M2.insert(i);
                if (i + 2 < n && s[i] == s[i + 2]) M3.insert(i);
        }
}

void update(int l, int r, int x) {
        fenadd(l, x);
        relax(l - 5, l + 5);
        fenadd(r, L - x);
        relax(r - 5, r + 5);
}

bool query(int l, int r) {
        auto it = M2.lower_bound(l);
        if (it != M2.end() && *it + 1 < r) return false;
        it = M3.lower_bound(l);
        if (it != M3.end() && *it + 2 < r) return false;
        return true;
}

signed main() {
        int t; cin >> t;
        while (t--) {
                cin >> n >> m >> s;
                build();
                while (m--) {
                        int tp, l, r; cin >> tp >> l >> r, --l;
                        if (tp == 1) {
                                int x; cin >> x;
                                update(l, r, x);
                        } else {
                                cout << (query(l, r) ? "YES" : "NO") << '\n';
                        }
                }
        }
}
```

Tutorial of Codeforces Round 903 (Div. 3)

+26 Vladosiya 3 weeks ago 122

Comments (122)

Write comment?

asdasdqwer

3 weeks ago, # | ☆ +12

I just have a question: Why is the algorithm for F correct? What's the intuition behind it?

→ Reply

Kareem_Elgoker

3 weeks ago, # ^ | ☆ 0

Its like diameter of the tree and you want to get a point in the middle of it you can say as a centroid

→ Reply

asdasdqwer

3 weeks ago, # ^ | ☆ 0

Yes, I know, but we essentially don't care about which nodes are marked and which ones are not.

→ Reply

robostac

3 weeks ago, # ^ | ☆ +2

Assume the tree is rooted at a marked node. If you are at the deepest marked node on a branch and move further down that branch it will increase the distances to all marked nodes by 1 (meaning the value of that node will be greater than the marked node you came from). You can therefore remove any nodes past the deepest marked node on each branch. From this it follows that you have a tree where all the leaves are marked and at that point it's the standard algorithm for finding the largest distance from a leaf.

→ Reply

JSR1

3 weeks ago, # ^ | ☆ 0

can't we do it with rerooting?

→ Reply

ohmygodtosbryantooorz

3 weeks ago, # ^ | ☆ ← Rev. 2 0

I actually did it with rerooting, with lazy segment tree.
You can check out my submission 227869909 .

→ Reply

T-Ahmed

3 weeks ago, # ^ | ☆ 0

There is no need for a lazy segment tree to do the rerooting.

→ Reply

chinesedfan

3 weeks ago, # ^ | ☆ ← Rev. 5 0

My rerooting with prefix and suffix sum. It is really a bit too more complicated, so I missed the first AK in div 3 (got AC after the contest 12min later, because of a little afraid of TLE by NodeJS).
https://codeforces.com/contest/1881/submission/227927645

The first DFS will calculate 3 kinds of values,

- dp[u], the max length from u to a marked node under u's subtree
- prefix[u][j], the max length from u to a marked node under u's [0,j] children's subtree
- suffix[u][j], the max length from u to a marked node under u's [j,] children's subtree

The second DFS is rerooting. For each u, its answer can be calculated by two parts,

- dp[u], to nodes under u's subtree
- `out` , to nodes not under u's subtree

When rerooting from `u` to `v` , `out` is updated by 3 parts. All of them should increase 1 if necessary.

- u's `out`
- u itself
- max(prefix[u][j-1], suffix[u][j+1]), with v is u's jth child

Hope above can help you improve understanding of

Hope above can help you improve understanding of rerooting, even this problem is not the best application to be solved.

→ Reply

3 weeks ago, # ^ | ☆ 0

Mradul_22

What is the time complexity of below code for problem D written by me..I'm little bit confused about the TC of my code..

include <bits/stdc++.h>

using namespace std;

define fastio() ios::sync_with_stdio(false),cin.tie(0),cout.tie(0)

//#To find min divisor of N int min_divisor(int n){
for(int i=2; i<=round(sqrt(n)); i++){ if(n%i==0) return i; } return n; }

int main(){ fastio();

```
int t;
cin>>t;

while(t--){

    int n;
    cin>>n;

    unordered_map<int,int> m;
    for(int i=0; i<n; i++){
        int x;
        cin>>x;
        int min_div = min_divisor(x);
        while(min_div!=x){              // what is the TC of this while
            m[min_div]++;               // is it simply O(logn) or O(logn*sqrt(n))
            x /= min_div;               // or something else please tell me!!
            min_div = min_divisor(x);
        }if(min_div!=1) m[min_div]++;
    }

    bool flag = true;
    for(auto &pr : m){
        if(pr.second%n!=0) flag = false;
    }
    cout<<(flag?"YES\n":"NO\n");
}
```

}

→ Reply

stefdasca

3 weeks ago, # ^ | ☆ 0

F can also be done in a similar fashion to Tree Distances —

Submission for Tree Distances

Solution for CF: 227871270

→ Reply

NyxNight

3 weeks ago, # ^ | ☆ 0

Exactly! Problem F can be considered as the special case of this one. 227869702 Here's mine with a slightly different approach.

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ 0

I mean this finding longest chain is quite common(at least i saw some of them before in div 3) Also I first found out the longest chain trick here(hard problem): https://oj.uz/problem/view/CEOI21_newspapers

→ Reply

Sahil_32

3 weeks ago, # | ☆ +17

"the author is lazy"

Pretty obvious from the way this editorial is written

→ Reply

helloworld1102

3 weeks ago, # | ☆ 0

For problem 1881F - Minimum Maximum Distance , can someone please tell the proof why it is always true to consider the longest distance between the labelled vertex and then divide it by 2 then take ceil of it.

After considering some examples it becomes somehow convincing but not very intutive about why that condition will always be true.

→ Reply

CppC_solver

3 weeks ago, # ^ | ☆ ← Rev. 2 +11

Define $dist(i,j)$ as the distance between node i and node j.

Assume that there are 3 labelled vertices a, b, c and (a, b) is the pair that creates longest distance from two labelled vertices. To find the node position that makes the minimum value of maximum distance to either a or b, ofc we choose the node (call this node d) that resides in the middle of the path from a to b (since if without loss of generality we assume $dist(d, a) \leq dist(d, b)$, then $dist(d, b) = \left\lceil \frac{dist(a,b)}{2} \right\rceil$ is the maximum from both and this is the min value of $dist(d, b)$).

Now, suppose $dist(d, c) > \max(dist(d, a), dist(d, b))$. Then, the longest distance from two labelled vertices must be the pair (c, a) or (c, b) (since $dist(d, c) + dist(d, b) > dist(d, a) + dist(d, b)$ and $dist(d, c) + dist(d, a) > dist(d, b) + dist(d, a)$) which results in contradiction.

→ Reply

helloworld1102

3 weeks ago, # ^ | ☆ 0

Thank you very much **CppC_solver** for your time and effort for clearing out the doubt.

→ Reply

123gjweq2

3 weeks ago, # | ☆ -12

C was the coolest one and it's not even close

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ +3

bruh I felt that G was much more interesting than all the problems(despite me trying 11 attempts to AC G)

→ Reply

Comsic

3 weeks ago, # | ☆ 0

In problem F how can we be sure that ceil(n/2) is the minimum answer compared to other nodes? Can someone please express their ideas on this?

→ Reply

CppC_solver

3 weeks ago, # ^ | ☆ 0

https://codeforces.com/blog/entry/121327#comment-1076496

→ Reply

Comsic

3 weeks ago, # ^ | ☆ 0

Thank you very much!

→ Reply

jeeva_24

3 weeks ago, # | ☆ 0

Starters were bad(ABC) but the main course was good(DEF) but i am not able to understand the bill(editorial)

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ +1

tru, i dont really enjoy ABC this round. DE was kinda meh for me. FG was quite interesting to play around with

→ Reply

Fluoresce

3 weeks ago, # | ☆ 0

Task D

If we assumed 1s can calculate 10^8 times, but in the worst condition(all a are equal to 999983), it will judge near 2000(t)*10^4(n)*10^3 times. Would it TLE?

I am a novice and thank you for your generous advice and point out my mistakes.

→ Reply

ohmygodtosbryantooorz

3 weeks ago, # ^ | ☆ 0

No, because the sum of n over all testcases does not exceed 10^4. So it's 10^4*10^3.

→ Reply

Fluoresce

3 weeks ago, # ^ | ☆ 0

oh,oh i misundertand it!thanks

→ Reply

Ahmed--Mohsen

3 weeks ago, # | ☆ ← Rev. 3 0

https://ideone.com/lyUPEZ

problem F c++ BFS code: why RTE on test 3 ???

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ ← Rev. 9 +3

Ok let me show some potential improvements of your code:

Why you are wrong: You missed out the corner case (n=1) where deg[1]=0 meaning your "start" is undefined and therefore shows runtime error. Meanwhile, the solution is wrong(after I tested it with the fixed code).

Data Structures used: You dont need to push the k values into a set instead use a boolean vector or smt. set requires O(nlogn) which means you will waste precious time if you are using set.

Your code checks if the root is in the set and the adjacent elements as well, you can generalise it by putting it after int node=q.front(); q.pop(); if(bla bla bla) mdis.push_back(...). You can save some lines of code.

You can try learning how to write functions such as function<void(int)> bfs=[&](int root){ code here}; this allows you to write the function inside your main fucntion where you dont need to pass all the arrays and stuff. You can access it inside the function. This should save you some implementation time as well.

Your code is not very clean (for a guy like me), therefore hard to read. I saw some parts of the code that has strange spacing. I used to be a template monster with around 400 lines of templates, but I find it hard to debug if I write a code with lots of shortforms(for some reason). I recommend spending sometime either looking at other people's code and learning a neater way of writing it. I'm not pinelizing you, just giving some personal suggestions

Neater and fixed code(although it is wrong): Sorry that I deleted the front part of your code.

code is here: https://ideone.com/fN9WEY

→ Reply

Ahmed--Mohsen

3 weeks ago, # ^ | ☆ 0

thank U for ur time your advice will be considered

→ Reply

Cosmoky

3 weeks ago, # | ☆ 0

Its a request for the author that do consider this point too that editorial is also referred by the beginners so they should make this editorial keeping in mind that beginners should also understand the intuition behind, making this platform truly helpful for the beginners as well!

→ Reply

Abracadaber

3 weeks ago, # | ☆ 0

I lost the top 150 due to hacking task D :(

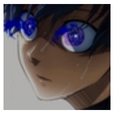
→ Reply

Omar_Mahfouz

3 weeks ago, # ^ | ☆ +4

same got TLe . because i am too dum dum and used frequency array instead of a map

→ Reply



Omar_Mahfouz

3 weeks ago, # ^ | ☆ 0

now instead of + 40 delta its a negative delta :(

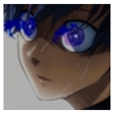
→ Reply

Abracadaber

3 weeks ago, # ^ | ☆ 0

the predictor showed me +140, and 136 was missing before blue, but now only +50. And it would also be my best result on codeforces

→ Reply



Omar_Mahfouz

3 weeks ago, # ^ | ☆ 0

sad :(now we learned something from this silly mistake at least.

→ Reply

Abracadaber

3 weeks ago, # ^ | ☆ 0

yep

→ Reply

Pirate_King

3 weeks ago, # ^ | ☆ 0

there's a predictor ?

→ Reply

Jexy

3 weeks ago, # ^ | ☆ 0

yes, there is a browser extension https://chrome.google.com/webstore/detail/cf-predictor/ocfloejijfhhkkdmheodbaanephbnfhn?utm_source=ext_app_menu

→ Reply

xboy

3 weeks ago, # ^ | ☆ 0

this is a better one imo

https://chrome.google.com/webstore/detail/carrot/gakohpplicjdhhfllilcjpfildodfnnn

→ Reply

Abracadaber

3 weeks ago, # ^ | ☆ 0

I didn't think about map either :(

→ Reply

chromate00

3 weeks ago, # ^ | ☆ +3

You don't really *need* the map. You just need a way to make the rollback/reset $O(1)$ amortized. This could be done by storing what was updated in a separate stack/queue/whatever.

→ Reply

Abracadaber

3 weeks ago, # ^ | ☆ 0

yes, but is it easier than just using map? In fact, if I had received a tle during the contest, I would have quickly rewritten everything on the map and what happened would not have happened.

→ Reply

→ Reply

chromate00

3 weeks ago, # ^ | ☆ 0

At least it's easier than rewriting, I should tell you. It's basically just `stk.push_back(k)` whenever you update index k, and `while(!stk.empty()){arr[stk.back()]=0;stk.pop_back()}` for the reset.

→ Reply

MostafaO4o

2 weeks ago, # ^ | ☆ ← Rev. 2 0

https://codeforces.com/contest/1881/submission/228412567 is this reset considered o(1)? as i didn't use stack to remember what i pushed and i didn't get TLE

→ Reply

chromate00

2 weeks ago, # ^ | ☆ 0

No, but the number of primes is not too big. Try finding out the value of $(\text{number of primes below } 10^5) \times 2000$

→ Reply

MostafaO4o

2 weeks ago, # ^ | ☆ 0

yea its O(T) right thank you :D

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ 0

map is good enough here lol

→ Reply

stefdasca

3 weeks ago, # | ☆ +3

Simple and insightful Video Editorials for A-E.

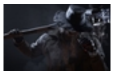
→ Reply

embetapbay

3 weeks ago, # | ☆ ← Rev. 4 0

Why is my code for incorrect? I feel like it closely resembles the intended idea. It's prob D. Please help me My code

→ Reply


VolodiaHeizenberg

3 weeks ago, # ^ | ☆ 0

I suppose in the end of the function "check" you should add max1 = max(max1, a) as it could be still the biggest prime number

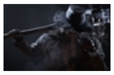
→ Reply

embetapbay

3 weeks ago, # ^ | ☆ 0

yes, i did it in my code.

→ Reply


VolodiaHeizenberg

3 weeks ago, # ^ | ☆ 0

Well, the everything I have done is just change array to map and it worked...

https://codeforces.com/contest/1881/submission/228020694

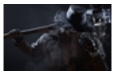
→ Reply

embetapbay

3 weeks ago, # ^ | ☆ 0

whyyyyyy ? :(

→ Reply


VolodiaHeizenberg

3 weeks ago, # ^ | ☆ 0

Well, I think I've found smth. I've made the array bigger and it managed to go through the test number 4. But now it gets TL on test 16...

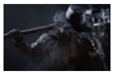
→ Reply

robostac

3 weeks ago, # ^ | ☆ ← Rev. 2 +9

memset doesn't work how you think (the size is in bytes, so you only reset 1/8th of the array). Having said that using the correct length will cause a TLE.

→ Reply

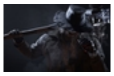


VolodiaHeizenberg

3 weeks ago, # ^ | ☆ 0

Well, interesting. So in this case should we just use vector instead of array?

→ Reply



VolodiaHeizenberg

3 weeks ago, # ^ | ☆ 0

Nah, it doesn't work too. Well, therefore is there any solution without using map? I take it there isn't.

→ Reply

embetapbay

3 weeks ago, # ^ | ☆ 0

227869532 i found jiangly's code.

→ Reply

MostafaO4o

2 weeks ago, # ^ | ☆ 0

i solved it without maps
https://codeforces.com/contest/1881/submission/228412567
but i hardcoded the prime count to 10power6 (i think i could counted it using seive)

can someone explain to me why map would be better?

→ Reply

embetapbay

2 weeks ago, # ^ | ☆ 0

because in 1 test case, you must check for 79000 primes. And with map, you simply need to check the prime factors. I think that

→ Reply

not_ahmed_hamed

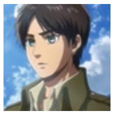
3 weeks ago, # | ☆ +7

For **F**, It's always better to re-construct the tree like this.

Tree

It's related to tree diameter (I guess there's no formal proof but this blog will help Blog)

→ Reply



Flash-

3 weeks ago, # | ☆ 0

Can anyone please explain how does D solution works? Can't able understand how counting frequency of prime divisors gives our answer.

→ Reply

myav

3 weeks ago, # ^ | ☆ ← Rev. 2 0

Let's use two ideas:

1) Every number can be represented as a product of prime numbers (= prime divisors). This is known as factorization.

2) If two numbers are equal, they have the same set of prime divisors.

So, the operation in the problem allows you to "take" one prime divisor from a[i] and "give" it to a[j].

If, after doing this operation several times, we can rearrange the divisors so that every number in the array has the same set of divisors, then the answer is YES.

To check that this is possible, we can simply count the number of occurrences of each divisor in the products.

For example, consider test case 4:

4

30 50 27 20

30 50 27 20

30 = 2 * 3 * 5

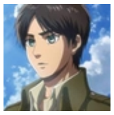
50 = 2 * 5 * 5

27 = 3 * 3 * 3

20 = 2 * 2 * 5

We have four occurrences of "5", four occurrences of "3" and four occurrences of "2". By doing the operation several times, we can turn each number into 2*3*5.

→ Reply



Flash-

3 weeks ago, # ^ | ☆ 0

Thanks! Understood

→ Reply

zwu2022015167

3 weeks ago, # ^ | ☆ 0

Damn!I wish I can saw your explaination earlier

→ Reply

kaikai508282737

3 weeks ago, # | ☆ 0

Only AC 1 problem... f**k me

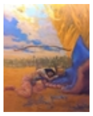
→ Reply

yl_neo

3 weeks ago, # ^ | ☆ +3

youll get better

→ Reply



noobiesAG

3 weeks ago, # | ☆ 0

If someone prefers video explanations or interested in knowing how to think in a live contest or reach to a particular solution.
Here is my live Screencast of solving problems [A -> E] (with detailed commentary in **Hindi** language).

→ Reply

not_ahmed_hamed

3 weeks ago, # | ☆ 0

Can **G** solved with sqrt-decomposition?, I'm getting WA2 Code

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ 0

I think it shoud be fine, WA means your code has some errors not TLE. But im pretty sure if your code has a large constant factor with sqrt-decomposition, your code might TLE

→ Reply

not_ahmed_hamed

3 weeks ago, # ^ | ☆ 0

Yup I know, I just wanted tye TLE verdict not the WA one to make sure that my idea is correct.

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ 0

Erm i also got many wa2(you can see from my submissions) I had many WA2 because I even pushed l into my valid answers when s[l-2]==s[l] but actually its invalid. If l is in the set meaning s[l]==s[l+2]. So check if your program has valid answers correctly. Also similarly for r with r+1 and r+2. r-1 with r+1.

→ Reply

not_ahmed_hamed

3 weeks ago, # ^ | ☆ 0

It's TLE now, the solution was right but I forgot a case $(i+1, i-1)$ when jumping by $sqrt$ size 228099350

→ Reply

yl_neo

3 weeks ago, # ^ | ☆ 0

Yeah I also missed the cases and kept on getting wa for like 10 times

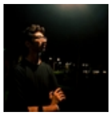
→ Reply

kar2003

3 weeks ago, # | ☆ 0

the editorial is short and to the point!!

→ Reply


sirvi_26

3 weeks ago, # | ☆ ← Rev. 2 0

.

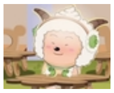
→ Reply

alaa.A

3 weeks ago, # | ☆ ← Rev. 3 0

Problem F solutions are based on the tree diameter, It takes some greedy proofs but in the end you can convert the problem to a direct tree diameter problem. Here's a blog

→ Reply


Luopei2022

3 weeks ago, # | ☆ 0

What does question C mean? please help me

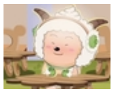
→ Reply

yl_neo

3 weeks ago, # ^ | ☆ 0

erm C is actually straightforward. Turning 90 degrees clockwise is not hard to understand. One operation actually just mean that you turn one cell's values to the next alphabet (a->b).

→ Reply


Luopei2022

3 weeks ago, # ^ | ☆ 0

I know What you mean. But I don't see example 4.Why 5?

→ Reply

Niteen77

3 weeks ago, # ^ | ☆ 0

so the test case is

baaa abba baba baab

now you can change a on the (1,4) cell to b

baab abba baba baab

now for the b on the (3,1) cell we will have to change the (1,2) , (2,4) and (4,3) cells to b

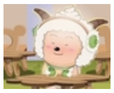
bbab abbb baba babb

lastly change the a at the (3,2) to b

bbab abbb bbba babb

so in total 5 operations

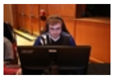
→ Reply


Luopei2022

3 weeks ago, # ^ | ☆ 0

I know all about it.Thank you very much.

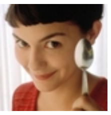
→ Reply


neetsk

3 weeks ago, # | ☆ 0

operation codeforces

→ Reply



3 weeks ago, # | ☆ 0

Any one Feels that this contest was not for Div.3 ?

It seems to be Harder :(

Folka

It seems to be Harder :(

→ Reply

ayush002024

3 weeks ago, # | ☆ 0

One more opportunity to be filled with shame.

→ Reply

brownfox2k6

3 weeks ago, # | ☆ 0

How do I solve 1881G - Anya and the Mysterious String with Segment tree using Lazy propogation? This is what I've done 227967016, I haven't used Lazy propogation and it's TLE. Thank you in advance.

→ Reply

yl_neo

2 weeks ago, # ^ | ☆ 0

I guess its pretty obvious but let me just repeat the obervations:

1. You only need to keep track of palindromes of size 2 and 3 only. pairs exist like this: (i,i+2), (i,i+1). Now since we do +x to every element in a range, we need to do range add(I used fenwick tree but lazy segment tree is also fine).
2. Hence we can store possible elements that has pairs (i,i+1), (i,i+2). Note : We store i not i+1, or i+2.
3. We only need to update (l-1,l) (l-1,l+1), (l-2,l), (r-1,r+1), (r,r+1) and (r,r+2) since the elements in range [l,r] is not changed.
4. Store the indexes in a set, binary search to find if right end<=r

Now you have finished the problem~~~

→ Reply

_bidhayak

3 weeks ago, # | ☆ 0

Why this code doesn't work for problem D?

code

→ Reply

chromate00

3 weeks ago, # ^ | ☆ 0

The product of the whole input is in the order of $O(A^n)$ which is at most 10^{60000} :unamused:

→ Reply

_bidhayak

3 weeks ago, # ^ | ☆ 0

No, product of every integer input per test case would be (10^10). correct me if I'm wrong.

→ Reply

chromate00

3 weeks ago, # ^ | ☆ 0

No man that's the sum not the product 💀💀💀

→ Reply

JustJie

2 weeks ago, # ^ | ☆ 0

LOL True

→ Reply

zwu2022015167

3 weeks ago, # | ☆ 0

Could someone explain me the E? How to think the dp which from right to left

→ Reply

yl_neo

2 weeks ago, # ^ | ☆ 0

Let us say we have block: 2 3 4 5 2 1 1

We can see that if we choose a[0], we continue with array 5 2 1 1

Meaning, its possible to make it into smaller subtasks: use dynammic programming.

We can see that if we want the minimum value for dp[0], we need to find dp[0+a[0]+1]=dp[3]

But we know, we cannot choose a block of 5. Hence, dp[3]=dp[4]+1.

Then we can see that either dp[i]=min(dp[i+1]+1,dp[i+a[i]+1])

Then we can see that either dp[i]=min(dp[i+1]+1,dp[i+a[i]+1])

Does that make sense? Ill explain further if that is not clear enough

⟶ Reply

MostafaO4o

2 weeks ago, # ^ | ☆ ← Rev. 2 0

i have a small question i used to solve dp by thinking **bottom up** and extending the problem (for example the subset sum problem [(Previous SUM)(Extended Number)] and now i can solve it by checking the previous state) however in this problem i failed to find a bottom up solution :(

you should start with the whole problem then break it down (Top Down Approach)

the question: are there certain problems which can be solved by only one of the 2 approaches?

⟶ Reply

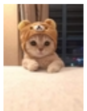
yl_neo

13 days ago, # ^ | ☆ 0

I actually understand dp as a breakingdown of subtasks, where you have the dp states as the stuff you need to record down for that each subtask.

Tbh, I actually dont really think in the form of top down or bottom up? I would just think how the subtask can come from. The number of states of the dp would be determined by yourself, so its all good as long as you define it right.

⟶ Reply



Ryukendo01

3 weeks ago, # | ☆ +4

I feel it was really a nice contest for anyone starting out on cp in the sense that it offers a good mix of adhoc and dsa

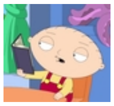
⟶ Reply

K_K_R_M

2 weeks ago, # ^ | ☆ 0

How to find adhoc problems? I really don't understand it.

⟶ Reply



Go8

3 weeks ago, # | ☆ +3

f can be solved using dp Let define d(v) as the max distance from vertex v to a labeled vertex in its subtree and define g(v) as the max distance from vertex v to a labeled vertex which is not in the subtree of v then d(v)=max(d(child of v)+1), g(v)=max(g(parent of v)+1,d(sibling of v) + 2) then f(v)=max(d(v),g(v))

⟶ Reply

adarsh.kumariitbhu

3 weeks ago, # ^ | ☆ 0

can u plz explain the intuition and the process as well as the implementation...plzz !!

⟶ Reply

TyrannosaurusRexx

3 weeks ago, # | ☆ ← Rev. 2 +3

F is similar to

Tree Distances I

The only change is to fix the end points of the diameter as RED nodes.

⟶ Reply

Jack_Daniels_

3 weeks ago, # | ☆ ← Rev. 2 0

Help please, why my solution has TL?(problem D) 228167450

⟶ Reply

JustJie

2 weeks ago, # | ☆ 0

I did rerooting for F, would've gotten top 100 or something but I had to have a typo that I didn't see

⟶ Reply

_oO_Oo_

2 weeks ago, # | ☆ 0

I have a question: why is such brutal decomposition with every integer working with Problem D It was my idea in the first place during the contest but I gave up cause I thought it would be to slow. So I took a binary search approach My code and resulted in a TLE.

Why is tutorial's code fast enough or why is my code too slow?(I learned after a better approach of

Why is tutorial's code fast enough or why is my code too slow? (I learned after a better approach of precomuputing the smallest prime decomposition of every number But still my original code confuses me.)

→ Reply

Aaditya9594

2 weeks ago, # | ☆ ← Rev. 6 0

C++ Solution for A 227837666

→ Reply

Aaditya9594

2 weeks ago, # | ☆ ← Rev. 2 0

C++ solution for B 227857937

→ Reply

Aaditya9594

2 weeks ago, # | ☆ ← Rev. 2 0

C++ solution for C 227893269

→ Reply

Aaditya9594

2 weeks ago, # | ☆ 0

C++ solution for D 228002421

→ Reply

Aaditya9594

2 weeks ago, # | ☆ 0

C++ solution for E 228148743

→ Reply

Aaditya9594

2 weeks ago, # | ☆ 0

C++ solution for F 228240931

→ Reply

sreejith

2 weeks ago, # | ☆ 0

In F, "Let's run a breadth-first traversal from any labeled vertex v1 and find the farthest other labeled vertex v2 from it.".

Why cant we take a "non marked" vertex v1 and find farthest marked v2 ?

→ Reply

sreejith

2 weeks ago, # ^ | ☆ 0

Even if we take non marked vertex also it's working. Not sure why the author intended to do like this ?

→ Reply

BABY_STEPS_

2 weeks ago, # | ☆ 0

Is there a more efficient way to solve A??

→ Reply

1021839

2 weeks ago, # | ☆ +1

I have two nicer(?) solutions than the official ones.

B: Suppose everything has length l in the end. Then $l \mid A, l \mid B, l \mid C$ (proof: run the splitting apart in reverse). This is also sufficient, and uses $\frac{A}{l} + \frac{B}{l} + \frac{C}{l} - 3$ operations. The number of operations is minimised when l is maximum, so $l := \gcd(A, B, C)$; to solve the problem just check if the minimum is at most 3. 228383697

(I saw a lot of people at the top of unofficial standings iterating over $(A + B + C)/l$ and testing each of them using this condition; it's quite strange to me that they didn't use GCD instead...)

G: There is no need for Fenwick trees or segment trees, you can check palindromes using just the difference array (and maintain bad positions using `std::set` as in the tutorial). 228383431

→ Reply

2 weeks ago, # | ☆ 0

Can someone Please help me with the problem F . My code 228434471 is giving wrong answer at Test 4.

→ Reply

hynmj

→ Reply

variety-jones

new, 3 days ago, # ^ | ☆ 0

Take a look at Ticket 17107 from **CF Stress** for a counter example.

→ Reply

shorya1835

2 weeks ago, # | ☆ 0

Problem G with sets and vectors is also possible submission

→ Reply

Entropy.

2 weeks ago, # | ☆ 0

Nice Contest, Thanks for the editorial.

→ Reply

WABot

2 weeks ago, # | ☆ 0

Can someone please explain the intuition behind problem F editorial? I knew it's right but I don't think next time I saw a similar problem I can just figure it it's about the tree diameter.

→ Reply

Kusoul

13 days ago, # | ☆ 0

Can anybody can tell me whether my solution in F can use when it have the edge-weight?Can it be called DP? 228766385

→ Reply

TyrannosaurusRexx

11 days ago, # | ☆ 0

Can someone explain what does this mean ~ `Notice that palindromes do not appear or disappear inside a segment, but they can appear or disappear at its boundaries`

→ Reply

mir183

new, 34 hours ago, # | ☆ 0

I did the same thing for A(1881A—Don't Try to Count), but I got hacked.

→ Reply