CODEFORCES β
Sponsored by Telegram

| shreyansh.barodiya | Logout

HOME CONTESTS GYM PROBLEMSET GROUPS RATING API RCC VK CUP HFT BATTLE

ADEDALIC BLOG TEAMS SUBMISSIONS GROUPS CONTESTS PROBLEMSETTING

adedalic's blog

Codeforces Round #421 Editorial

By **adedalic**, history, 10 days ago,

First, I apologize for problems with round and serious problem with Div1A/Div2C. It was very important round for me and, as always, something goes wrong. **KAN** have already wrote about this.

Anyway, there are problems, so editorial must exists. Due some circumstances, Editorial will be upload in parts. Also, most of tutorials will contain main ideas how to solve task, not ready algorithm.

820A - Mister B and Book Reading

All that needed - is to accurately simulate process.

Create variable, which will contain count of read pages, subtract l, add v_0, check, what you still have unread pages, make $v_0 = min(v_1, v_0 + a)$ and again.

Complexity is $O(c)$.

code

820B - Mister B and Angle in Polygon

Since polygon is regular, all vertices of a regular polygon lie on a common circle (the circumscribed circle), so all possible angles are inscribed angles. And all inscribed angles subtending the same arc have same measure.

More over, minor and major arcs between vertices v_i and v_k equals to minor and major arcs between vertices v_{i+1} and v_{k+1}.

And finally, length of arc can be calculated with formula as sum of minor arcs between consecutive vertices. Length of minor arcs between consecutive vertices equals to $360 / n$.

Length of inscribed angle is half of arc it based on.

In other words $\angle v_1 v_2 v_3 = \frac{180 \cdot (n - (v_3 - v_1))}{n}$ if $v_1 < v_2 < v_3$ or $\frac{180 \cdot (v_3 - v_1)}{n}$ in other case.

In the end, this task can be solved by checking all different $(v_3 - v_1)$ (v_1 can be fixed as 1), or by formula, if we put in $|\angle v_1 v_2 v_3 - a|$ formula above.

In result finding closest angle can be done in $O(n)$ or even $O(1)$ time.

code

819A - Mister B and Boring Game

Tutorial is not available

code

819B - Mister B and PR Shifts

Let's see, how p_k $(1 \le k \le n)$ affects different shifts.

Let's denote d_i is deviation of the i - th shift. At first all $d_i = 0$.

Then p_k affects it in following way:

→ Pay attention

Before contest
VK Cup 2017 - Finals (practice session)
14:21:12

Like 63 people like this. Be the first of your friends.

→ shreyansh.barodiya

Rating: 1355
Contribution: 0

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

shreyansh.barodiya

→ Top rated

#	User	Rating
1	tourist	3602
2	Petr	3265
3	Um_nik	3183
4	ainta	3174
5	Radewoosh	3163
6	W4yneb0t	3148
7	LHiC	3139
8	izrak	3109
9	RomaWhite	3053
10	moejy0viiiiiv	3051

Countries | Cities | Organizations View all →

→ Top contributors

#	User	Contrib.
1	rng_58	180
2	Errichto	168
3	csacademy	164
4	Petr	160
5	tourist	158
6	Swistakk	155
7	Zlobober	150
8	zscoder	145
9	GlebsHP	142
10	matthew99	141

View all →

→ Find user

Handle:

- $d_0 += |p_k - k|$,
- $d_1 += |p_k - (k + 1)|$,
- ...
- $d_{n-k} += |p_k - n|$,
- $d_{n-k+1} += |p_k - 1|$,
- ...
- $d_{n-1} += |p_k - (k - 1)|$.

Then there are 2 cases: $p_k \geq k$ or not.

If $p_k \geq k$ after removing modules we will get 3 query:

- to add $p_k - (k + i)$ to d_i where $0 \leq i \leq p_k - k$,
- to add $(k + i) - p_k$ to d_i where $p_k - k < i \leq n - k$ and
- to add $p_k - i$ to d_{n-k+i} where $0 < i < k$.

Else if $p_k < k$ we need to perform next operation:

- to add $(k + i) - p_k$ to d_i where $0 \leq i \leq n - k$,
- to add $p_k - i$ to d_{n-k+i} where $1 \leq i \leq p_k$ and
- to add $i - p_k$ to d_{n-k+i} where $p_k < i < k$.

But in both cases we must add 3 arithmetic progression to the segment of array d. Or make operation of adding $k \cdot (x - l) + b$ to segment $[l, r]$. Its known task, which can be done by adding/subtracting values in start and end of segment offline.

To make such operation we need to remember, how to add value b to segment $[l, r]$ of array d offline. We can just do next operations: $d[l] += b$ and $d[r + 1] -= b$. Now value in position i $ans_i = \sum_{j=0}^{j=i} d[j]$.

So what is adding progression with coef k? it's only adding to array d value k to all positions in segment $[l, r]$. That's why we need other array, for example df and making $df[l] += k$ and $df[r + 1] -= k$. In result, $d[i] = d[i] + \sum_{j=0}^{j=i-1} df[j]$.

So algorithm to add $k \cdot (x - l) + b$ to segment $[l, r]$ is next:

- $d[l] += b$,
- $d[r + 1] -= k \cdot (r - l + 1)$,
- $df[l] += k$,
- $df[r + 1] -= k$.

After all queries we need recover array d with formula $d[i] = d[i] + \sum_{j=0}^{j=i-1} df[j]$. And after that get answer with formula $ans_i = \sum_{j=0}^{j=i} d[j]$.

So complexity is $O(n)$.

code

819C - Mister B and Beacons on Field

There 2 stages in this task: moving of first beacon and moving of second.

But at first we need factorization of n and s. Since n and s are product of integers $\leq 10^6$, it can be done in $O(log(n) + log(s))$ time by "Sieve of Eratosthenes".

Start from **second stage**, when second beacon is moving:

Position of beacons will look like pair of points: $(0, 0)$, $(0, k)$, where $0 \leq k \leq n$.

We need to check existing of point (x, y) such, that area of triangle $(0, 0)$, $(0, k)$, (x, y) equals to s. Using cross product $|((0, k) - (0, 0)) \cdot ((x, y) - (0, 0))| = 2 \cdot s$. After simplifying we get $|k \cdot x| = 2 \cdot s$ where $0 \leq k \leq n$.

So we can iterate all divisors of $2 \cdot s$, using factorization of s and recursive algorithm.

Complexity of second stage is $O(\sigma(s))$, where $\sigma(s)$ — number of divisors of s and for $s \leq 10^{18}$ $\sigma(s) \leq \approx 10^5$.

In the **first stage** we have such points: $(k, 0)$, $(0, n)$, where $1 \leq k \leq m$.

Find

→ Recent actions

csacademy → Balkan OI 2017 Day 2 online mirror

FallAndRise → Problem on combinatorics

Hasan0540 → [GYM] 2017 PSUT Coding Marathon

kieuquocdatcvp → Need help with a tree problem in SCPC 2017

yeguanghao → Internet Problem Solving Contest 2017 Registration Is Open Now!

bacs.official → BACS Regional Programming Camps 2017 Selection Contest

vamaddur → Comparing Permutations of Subarrays Problem

Florin29 → Cannot get AC on Mashmokh and Reverse Operation, test #72 output is wrong

mayankpathak → Sorting array element by frequency in O(nlgn) time

gotop → Compute "distribute n candies to to r people"

zemen → IndiaHacks 2017

vamaddur → Can someone explain this solution to USACO 2013 Gold December Contest: The Bessie Shuffle?

casimir → Why does codeforces not use https?

kursatbakis0 → I have been coding for 5 years and I am giving up

PraveenDhinwa → Invitation For July Long Challenge 2017

arkapravapaul → Problem CRYPTO1

vamaddur → WA in Finding the Minimum Vertex Cover of a Tree with DP

darkshadows → Insightful tutorial on Centroid Decomposition of a Tree

xuanquang1999 → Need help with another graph theory problem

removed1 → Codeforces Beta Round #1 - Tutorial

BledDest → Educational Codeforces Round 24

slow_coder4 → Help! set stl

Xellos → Invitation to IPSC 2017

warrior98 → Minimum-cost cycle

Rhangel → How to find a k-th bigger element in sequence

Detailed →

We need to check existing of point (x, y) such, that area of triangle $(k, 0)$, $(0, n)$, (x, y) equals to s. Using cross product $|((k, 0) - (0, n))\cdot((x, y) - (0, n))| = 2\cdot s$ we can get next equation: $|k\cdot(y - n) + n\cdot x| = 2\cdot s$. Then solution exists iff $gcd(k, n) \mid 2\cdot s$ $(2s \% gcd(k, n) = 0)$.

And we need to calculate how many $1 \le k \le m$ such, that $gcd(k, n) \mid 2\cdot s$.

We will solve it in next way: let's $n = p_1^{n_1}p_2^{n_2}... p_l^{n_l}$ and $2s = p_1^{s_1}p_2^{s_2}... p_l^{s_l}$ $(n_i, s_i \ge 0)$.

Look at all p_i, that $n_i > s_i$. It's obvious, that if $p_i^{s_i + 1}$ is divisor of k, then $2s$ doesn't divide at $gcd(k, n)$.

In result, we have some constrains on k, like k doesn't divide at $a_j = p_{i_j}^{s_{i_j} + 1}$.

Finally, we have next task: calculate number of k $(1 \le k \le n)$ such, that k doesn't divide any of a_i. It can be done with inclusion-exclusion principle, where number of k which divides a_{i_1}, a_{i_2} ... a_{i_b} is $\frac{n}{a_{i_1}a_{i_2}...a_{i_b}}$.

Complexity of first stage is $O(2^{z(n)})$, where $z(x)$ — number of prime divisors of x, $z \le 15$ for integers up to 10^{18}.

Result complexity is $O(\sigma(s) + 2^{z(n)})$ per test.

code

819D - Mister B and Astronomers

Let's construct slow but clear solution and then, speed it up.

Let's denote $s = \sum_{i=1}^{i=n} a_i$. We can see, that, at first, all operation with time are modulo T and the i - th astronomer checks moments st_i, $(st_i + s)\%T$, $(st_i + 2s)\%T$ ..., where $st_i = \sum_{j=2}^{j=i} a_i$. More over, every astronomer, who checks moment t will check moment $(t + S)\%T$ by next time.

So we now constructed functional graph with T vertices. But this graph has very special type, since it can be divided on some cycles. More specifically, This graph consists of $gcd(T, s)$ oriented cycles and each cycle has length exactly $\frac{T}{gcd(T,s)}$. Even more, vertices u and v belong to same cycle iff $u \equiv v\ (mod\ gcd(T, s))$.

So we can work with cycles independently.

Let's look closely on one cycle. It's obviously, that all astronomers will walk on their cycles from their starting positions $st_i \% T$. But what the answer for them.

Answer for the i - th astronomer is number of vertices, including starting vertex, to the nearest starting vertex of any other astronomer if count along the orientation of cycle, because if two astronomers came to same vertex, lucky is one, who came first. Other astronomer has this vertex as start, his time is st_j, i - th time is $st_i + k\cdot s$, $k \ge 1$ and $st_i + k\cdot s > st_j$. If $st_i \equiv st_j\ (mod\ T)$ and $st_i < st_j$ then answer for the j - th astronomer is always 0.

So we must effectively calculate distance between two positions on cycle.

For that, let's numerate vertices along the orientation of cycle using vertex with minimal label as 0. If we will know position of $st_i \% T$ for every astronomer calculation of distance between consecutive is trivial (sort, or set or other).

For the i - th astronomer let's denote vertex with label 0 in his cycle as z_i. $z_i = st_i \% gcd(T, s)$. But cycles very specific, because vertex with label 0 is z_i, vertex with label 1 is $(z_i + s) \% T$, vertex with label 2 is $(z_i + 2s) \% T$. In other words, vertex with label k is $(z_i + k\cdot s) \% T$.

If we want to know position k of st_i, we need to find v such, that $v \equiv (v \% gcd(T, s)) + k\cdot s (mod\ T)$ which is diofant equation and can be calculated in $O(log(T))$ time.

Result complexity is $O(n\cdot log(T))$.

code

819E - Mister B and Flight to the Moon

There are different constructive solutions in this problem. Here is one of them.

Consider odd and even n separately. Let n be even. Let's build for each even $n \geq 4$ a solution such that there are triangles `1-2-x`, `3-4-y`, `5-6-z` and so on. For $n = 4$ it's easy to construct such a solution. Then let's add two vertices at a time: $a = n - 1$ and $b = n$. Instead of triangle `1-2-x` let's add triangle `1-a-2-x`, square `1-a-2-b` and square `1-2-b`. The same with `3-4-y`, `5-6-z` and so on. Only one edge `a-b` is remaining, we should add it twice. To do this let's replace the square `1-a-2-b` with triangles `a-b-1` and `a-b-2`. Easy to see that the condition on triangles is satisfied, so we can proceed to adding two more vertices and so on.

To deal with odd n let's keep triangles `2-3-x`, `4-5-y` and so on. To add two more vertices replace each of these triangles with two squares and one triangle in the same way as for even n, and also add two triangles `1-a-b`.

code

Tutorial of Codeforces Round #421 (Div. 1)

Tutorial of Codeforces Round #421 (Div. 2)

+205 adedalic 10 days ago 33

Comments (33)

Write comment?

adedalic

10 days ago, # | +12

Auto comment: topic has been updated by ***adedalic*** *(previous revision, new revision, compare).*

→ Reply

ibrahim5253

10 days ago, # | +8

But in both cases we must add 3 arithmetic progression to the segment of array d. Its well known task, which can be done by adding/subtracting values in start and end of segment offline.

Can you please elaborate a bit(or point to some reference) on this *well known task*?

→ Reply

Hailo

10 days ago, # ^ | ← Rev. 2 0

I a general case when you need to add progressions and answer querys you can use a Lazy Fenwick tree, see this tutorial. Maybe someone knows how yo solve it whitout this structure, I don't.

But in this specific case, you made updates but you have to answer the query just after all modifications are made. So you can made the updates using just 2 arrays

→ Reply

adedalic

10 days ago, # ^ | +5

Yes, i will elaborate this with next update

→ Reply

8shubham

10 days ago, # ^ | +1

Reference for offline addition(407C)

→ Reply

forerunner

10 days ago, # | ← Rev. 4 +6

A"if...else if"solution for div2 C... 28115691

→ Reply

shreyash390

10 days ago, # ^ | ☆ +6

can you please explain your code??

→ Reply

forerunner

9 days ago, # ^ | ☆ 0

sorry that I failed to prove it when b < a,seems complex...I just think it will be a loop; and when b >= a or the length is short,the answer is sure.there are some mistakes before,now corrected and some comments added

→ Reply

ranjanvittal

9 days ago, # ^ | ☆ +1

A loop come brute force solution must work I guess. Basically logic I tried out is checking all strings where the repeating character introduced by B is any of the last max(1, a-b) characters.

http://codeforces.com/contest/820/submission/28126243

→ Reply

meeeep

9 days ago, # | ☆ +5

An alternative to handle the even case of 1E is to split the graph into 2 equal disjoint sets A and B. Label the elements a_1 to a_m, b_1 to b_m.

Use induction to construct for A and B separately. What remains is the complete bipartite graph, which can be done using cycles of length 4, all of form a_i — b_(i+j) — a_(i+1) — b_(i+j+1), where i and j run from 1 to m, and indices are taken modulo m.

→ Reply

Aronzx

9 days ago, # | ☆ ← Rev. 2 0

Can someone please help me understand how Div2 Problem B can be solved in O(1) ?

→ Reply

Sina_Imani

9 days ago, # ^ | ☆ 0

Here's my submission Well, if you want to find closest angle to a, you just need to find an arc with angle most close to 2*a which its vertices are vertices of polygon. For doing this, you make your ideal arc (with angle 2*a) and find the vertex in polygon which is closest to end of arc.

→ Reply

joker_in

9 days ago, # | ☆ +4

can anyone explain div2d div1 b more thoroughly . i am not getting it

→ Reply

usernameson

9 days ago, # | ☆ +3

The solution for Div 2 B is really nice. My solution was to construct the polygon then use dot products and a binary search to find the best end point with two other points fixed. It was really tedious and time consuming to code and had a worse complexity of $O(n \log(n))$.

→ Reply

HossamDoma

9 days ago, # | ☆ 0

Can someone elaborate more accurately on Div2 D?

→ Reply

haleyk100198

9 days ago, # | ☆ +18

For Div2D / Div1B, one could also keep the amount of elements which holds a[i] >= i to update the difference in O(1) time.

Code with comments : http://codeforces.com/contest/819/submission/28113390
→ Reply


end83

9 days ago, # ^ | ☆ 0

Can you explain your solution a bit more?
→ Reply

haleyk100198

9 days ago, # ^ | ☆ +7

To calculate the weighted sum of the (i+1)-th shift from the i-th shift, the elements which holds i >= a[i] (named as gt in the code) contributes +1 to the sum and i < ai contributes -1. That being said, we could solve the problem by simply maintaining the amount of elements which holds true on the above cases.

For non-tail elements, we could easily tell the moment of i < a[i] becomes i >= a[i] is a[i] — i, meaning that we need to account for this moments later.

For the tail element at the moment, we shall recalculate its contribution and placing a new update for it. Note that as a[i] <= n, it is guaranteed that n > a[tail] holds true, so we shall remove one from gt before the update. Same for updating lt for the tail.
→ Reply


R_ash

9 days ago, # ^ | ☆ 0

Can you please tell me what does upd[i] hold? Actually I don't undersatand what this operation "upd[a[i]+i]++" is doing.
→ Reply

haleyk100198

8 days ago, # ^ | ☆ +1

As upd[time] stores the pended update for moment "time", upd[time]++ means that there will be one extra element will switch from contributing -1 to +1. As we are placing a[tail] back to the front, after a[tail] iterations it will hit a[index] == index, therefore we shall place an update on time = a[tail] + current_time = a[tail] + i.
→ Reply


nikich340

8 days ago, # ^ | ☆ 0

I did it more stupid using segment tree, but calculating indexes carefully is a bit difficult, and each update in log time :P

http://codeforces.com/contest/820/submission/28132944
→ Reply


s_jindal00

7 days ago, # ^ | ☆ 0

Awesome solution! Thanks. I understood it and coded myslef but it's giving TLE! I dont understand how a O(n) solution gets TLE?! Code
Can you please check it once why its happening
→ Reply

haleyk100198

7 days ago, # ^ | ☆ +1

Editted version

I interchanged line 10 & 11 of the original code and it passed all cases, the TLE is most likely caused by initializing the array with size n+2 before n is properly initialized.

For the sake of competitve programming, I would recommend you to use array that has fixed size instead of referring their sizes to variables -- This increases memory usage in practical cases but we only care about the worse case scenario here.
→ Reply


s_jindal00

7 days ago, # ^ | ☆ 0

Yea. I figured it out just after submitting the solution in java. In c++ uninitialized variables contain garbage values and no compilation error, however in java you need initialise it. **BTW thanks for giving this nice concept to solve the problem**

→ Reply


madn

9 days ago, # | ☆ ← Rev. 2 +11

I think my solution of div2D/div1B is a bit easier. We can see that after a cyclic shift our array is always divided into two parts: shifted part and not shifted part. So i just simulated the process. All wee need is two maintain two values for each of two parts: number of such elements that |p[i] — i| will increase after shifting, and number of elements for which this value will decrease. Complexity is O(n). My explanation isn't very good, but code may help you 28121452

→ Reply


arman.t

9 days ago, # ^ | ☆ 0

I did something similar: http://codeforces.com/contest/820/submission/28127444

→ Reply


adedalic

9 days ago, # | ☆ 0

Auto comment: topic has been updated by ***adedalic*** *(previous revision, new revision, compare).*

→ Reply


cdkrot

9 days ago, # | ☆ ← Rev. 2 +38

Abstract.

Another (and probably more simple) approach to Div1E.

Here I will suggest a way of constructing a solution for n, given **any** solution for $n - 2$. No properties are requested and required to be preserved by such induction.

You still need to solve cases for $n = 3$ and $n = 4$ to get the full solution.

The method:

Let's take two nodes, let them be $s = n - 1$ and $t = n$ for simplicity of numbering.

Consider following paths:

- $s \to 1 \to t$
- $s \to 2 \to t$
- ...
- $s \to (n-2) \to t$
- $s \to t$ (this one is special from above).

Taking any two of such pathes you form a valid cycle of length 3 or 4. Match listed paths into cycles the following way: first with second, second with third, ..., pre-last with last, and last with first (in other words, match by neighbourhood).

Add such cycles to the answer. Note, that each listed path was used twice by construction above, this way all edges connected to s or t were used twice too.

Continue with $n = n - 2$ here.

→ Reply


geniucos

7 days ago, # ^ | ☆ 0

This solution is absolutely brilliant! Thanks for sharing it with us. It's unexpectedly elegant and easy to understand. I've looked for an induction approach but failed to think about it for long because it was hard for me to imagine that I might be able to keep the old construction in its exact form, without some strange alterations.

→ Reply

joker_in

8 days ago, # | ☆ 0

why editorial link is not there on contest page . I know there is issue with div2-c question but rest of the question are good , i learned and solved div2-d and e(almost done) . good work in editorial and contest **adedalic** . thumbs up for good work.

→ Reply

XuYipei

7 days ago, # | ☆ ← Rev. 2 0

There is my solution to 819A - Mister B and Boring Game First of all, I guess the color for each segment a is ascending or descending. And for each segment b its color is the minimum or maximum value of the previous paragraph a. We can easily make the interval between l and r no more then 5*(a+b). Thus, you can enumerate the status of each segment and calculate the current answer. Maybe I can't describe it clearly.This is my submission:28097355

→ Reply

minson123

7 days ago, # ^ | ☆ 0

The description said: "From multiple variants of t lexicographically minimal is chosen." Why each segment a is descending?

→ Reply

7 days ago, # ^ | ☆ 0

Your solution inspire me.

minson123

This is my solution.

For each segment a is increasing then as you said: "for each segment b its color is the minimum or maximum value of the previous paragraph a.", so I separate it to two situation: If a<=b then get the minimum/maximum value in section[lst+1,lst+a] else get it in section[lst+b+1,lst+a].

→ Reply

Codeforces (c) Copyright 2010-2017 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jul/07/2017 23:07:21 UTC+5.5 (c3).
Desktop version, switch to mobile version.
Privacy Policy