CodeChef Discussion

How to solve EQUALMOD from yesterday's contest

1

Can anyone explain their solution for EQUALMOD problem which was asked in ICPC online round 2017 ?

help acm-icpc

converted to question 06 Nov '17, 18:53
4★ vijju123 ♦♦
[15.3k]●1●20●62

asked 06 Nov '17, 18:42
4★ gitesh18
[33]●6
accept rate: 0%

1

3 Answers:

oldest answers newest answers popular answers

4

I was not able to submit my solution during the contest (spent a lot of time on Compression Algorithm), so, I don't know if my approach is correct. Please let me know if you find a mistake in this.

The first thing that you need to see is that since you need to make all (ai mod bi) equal, the value that you will finally get after performing all the increment operations, will never be greater than b_min - 1. So, we need to set all the modulo to some number between 0 and b_min - 1. The obvious approach is to find the minimum cost to set the modulo to each number in this range, and simply print the minimum.

First, lets see how to calculate the minimum cost to set all of the modulos to some particular value, say, 'x'.Since the only operation allowed is to increment, for all those positions having (ai mod bi) < x, you need to increment them until they reach 'x'. For all those positions that have ai mod bi > x you need to make them reach zero first, and then increment them 'x' times. You can find this cost quickly if you maintain prefix sums for (ai mod bi) and suffix sums for (bi - ai mod bi) after sorting based on (ai mod bi).

Basically, every 'x' will split the sorted array into two halves, say, left and right and we know how to find the cost, given this splitting point. Now, the 'x' that you're trying can be of two types, it can either be equal to one of the modulos already present in the array or not. There will be 'n' values for 'x' at most, if 'x' is equal to one of them, we can try each of those and store the minimum.

But, what if 'x' is not equal to any number in the array ? Then, 'x' will lie in between two adjacent numbers in the array. I will explain this with an example:

A : {1, 2, 10, 20, 25}

B : {2, 4, 12, 22, 30}

Ai % Bi : {1, 2, 10, 20, 25}

Suppose that you pick 'x' and it lies in the interval between 10 and 20, say, x=11. Let's see what will the cost be to set all the values equal to 11. There are 3 numbers less than 11 and 2 numbers greater than 11. You can see that the cost for the left part ie. the first 3 numbers will simply be 11*3-(1+2+10) and the cost for the right part will be 2+5 + 11*2. What if you now take x = 12 ? your cost wil simply increase by 1 for each number in the array. This will happen each time you take a step to the right from 11 ie., 12,13,14,.... So, its obvious that for 'x' lying in an interval between two numbers in the array, the best possible value will be the first number in the interval.

There will be N-1 intervals like this lying in between two numbers in the array, and two more, one before the smallest and after the largest, which you can easily handle separately. The answer will the minimum of these, and the minimum that we stored previously by taking 'x' as each of the numbers in the array. Let me know in the comments if there's something that you don't understand, or if you find some mistake in my approach.

link | award points

answered 06 Nov '17, 22:52
6★ hemanth_1
[1.4k]●11
accept rate: 28%

4

Here's a weird looking graph. On the X-axis are the elements of the array. Each element has its own graph. On the Y-axis are the choices of final value which every $A[i]$ should equal in the end. Whenever I say $A[i]$, assume I mean $A[i] \bmod B[i]$. The width at height x of the the graph of i^{th} element represents the cost of changing $A[i]$ to x.

Follow this question
By Email:
You are not subscribed to this question.
subscribe me
(you can adjust your notification settings on your profile)
By RSS:
Answers
Answers and Comments

Question tags:

help ×2,709

acm-icpc ×1,105

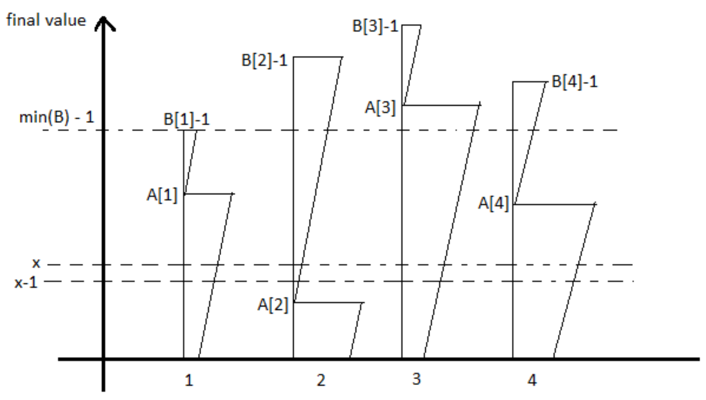
question asked: **06 Nov '17, 18:42**

question was seen: **1,694 times**

last updated: **07 Nov '17, 22:53**

Related questions

- Can anyone help me with A Few Laughing Men from ACM-ICPC Asia-Amritapuri Onsite Replay Contest...
- Please help me in this past ICPC problem
- Please help me solving Colorful Grids (ICPC16E)
- Please help me in ZUBGOLD from Kolkata Onsite.
- Can any one Explain this easy Question-Candy Game (Practice icpc) with Solution?
- INTEST: Getting rte
- Problem with codechef solution visiblity.
- Need help on how to start solving difficult questions
- [closed] Object problem in Java
- Enormous Input Test (INTEST)



Clearly the final value must be in $[0..\min(B)-1]$, so we just look at that range. Let $cost(x)$ be the cost of setting final value to x. Now it's clear that $x-1$ is a better choice of final value than x, since at $x-1$ each element will have 1 less cost. So $cost(x-1) = cost(x) - N$. If we keep decreasing by 1, this trend continues until it hits some $A[i]$. After that it jumps from $cost(A[i])$ to $cost(A[i]) - N + B[i]$. Again it will keep decreasing as before. So $cost(x)$ is a function with multiple local minima at each $A[i]$. There is also one final minima at 0. Once these minima are sorted in ascending/descending order, one minima can be obtained from the previous in constant time (difference$\times N+$adjust for $B[i]$ spikes). So just linearly iterate and find the global minimum :)

link | award points edited 07 Nov '17, 22:53 answered 07 Nov '17, 22:01
6★ meooow ♦
[7.1k]●7●18
accept rate: 48%

1

My team followed the exact same approach as mentioned by hemanth_1, but we got WA. However, post-contest we realized that since x is bounded by the range [0...b_min-1] , so taking all intervals from the array was possibly giving us WA. Instead, we should proceed for all intervals such that the first element of that interval is in the range [0.b_min-1].
This was a very good problem.
Would verify the approach once the problems are uploaded in the practice section.

link | award points answered 07 Nov '17, 12:35
5★ arunava5
[51]●1
accept rate: 0%

Yeah, I forgot to mention that..the first element must be in the range...I realised one more thing just now...the number to the left of every interval, ie. the number in the array, should give a better answer than the first number in the interval..So, I was thinking that its enough to simply check by taking 'x' equal to each element in the array less than b_min.

6★ hemanth_1 (07 Nov '17, 18:45)

Your answer

[hide preview] community wiki:

Preview

Answer the question