Editorial for CUET Intra University Junior Programming Contest: Round 4

A. Ichigo Eating Waffles!

Author: Asad **Ban.kai** Bin Saber

As Ichigo can shape waffles of any size, he will add all the waffles together to get a single big waffle.Then he will divide it by x, the standard waffle size, to get the maximum number of whole waffles he can get.

Ans = $floor(\frac{\sum_{i=1}^{n} ara_i}{x})$

Complexity: O(n)
Code: https://pastebin.com/612t1zG3

B. No Ragging

Author: Tanzim **Tanzim_bn** Bin Nasir

Firstly, if X<0, we can redefine X as −X, which just reverses all the directions of movement and makes it easier to approach the problem.

If you go straight from X to the origin and do not reach 0 ($X \geq K * D$), then $X - K * D$ is clearly the answer.

If not, it is obviously best to repeat the round trip to cross the origin. However, K is so messy that you can't loop naively. Well, if you make two round trips, you return to the same position, so you only need to consider the remaining number of steps mod 2.

Complexity : O(1)
Code : https://ideone.com/owoLcS

C. Need More Cuet-Bus

Author: Shams **-Rafid-** Ahnaf Rafid

To find out the minimum bus increment, we will do **binary search** on bus increment. In binary search, we will check from the given information if the bus increment number satisfy the condition (at least half of the total students are happy with this bus increments).

Complexity Analysis : **O(nlog(L))**. Here n is the number of departments and L is the maximum number of bus increment from the given information (because the answer won't be greater than the maximum bus increment from the given information).

Solution code : https://pastebin.com/0nvnagj1
Solution code (without binary search) : https://pastebin.com/6CtgCrbD

D. Save Friendship!!!

Author: Tutul **Tutul_dhar** Dhar

Solution Approach:

1. **Calculate Prefix XOR:** We first compute the prefix XOR of the array, which allows us to quickly compute the XOR of any subarray.
2. **Sliding Window XOR:** Using the prefix XOR, compute the XOR for every subarray of length m.
3. **Find Maximum by Flipping Bits:** For each XOR result, flip each bit up to the MSB to find the maximum possible value.
4. If we can not get any such segment of XOR value greater than 0 then simply output -1.

Complexity: O(n)
Author's Code: https://ideone.com/rmQOmy

E. BitMasters Challenge

Author: Ashfaqur **adibur6** Rahman Adib

In this bit manipulation problem, we'll analyze it bit by bit. First, let's determine the values of A_i ,B_i ,and C_i, where A_i represents the ith bit of A in binary.

Now, consider two cases for the ith value of g = XOR(A, B, C):

1. **If the ith value of g is 1:**
   XOR(A_i ,B_i ,C_i) equals 1 if either all of them are 1 or only one of them is 1. The input is valid if the sum of AND(A_i ,B_i), AND(A_i ,C_i), and AND(B_i ,C_i) is either 3 or 0.
   - If the sum is 3, then all the values of A_i ,B_i and C_i are 1.
   - If the sum is 0, then only one of them is 1
2. **If the ith value of g is 0:**
   XOR(A_i ,B_i ,C_i) equals 0 if either all of them are 0 or two of them are 1. The input is valid if the sum of AND(A_i ,B_i), AND(A_i ,C_i), and AND(B_i ,C_i) is either 0 or 1.
   - If the sum is 0, then all of them are 0.
   - If the sum is 1, then the pairs for which AND(X, Y) is 1 contain 1 in the ith bit.

Complexity : $O(log(x))$
Code: https://ideone.com/rirvjm

F. Again Prime

Author: Gourab **GOURAB** Biswas

There are given a range A to B . The range contains at most 21 prime numbers. You need to count the sum of the subset which is prime by using the prime numbers in the range A to B. Also the A,B value is maximum 10^5. So you can use sieve of eratosthenes to know if a number is prime or not. The array size of sieve can be 21*(10^5) = 2000000. Now you can take the prime numbers of the range A to B and use Bitmask to generate all possible subset sum and count when the sum is prime. To check prime use the array.

Complexity: $O(2^x)$, x = number of primes in [A, B].
Code: https://pastebin.com/EsV9TvB0

G. Ichigo and Part Time Job

Author: Asad **Ban.kai** Bin Saber

We will pre-calculate a partial answer for each node.

For this, for each node the value will be equal to: 2(to reach a child node from the current node and come back to that current node) + answer for that child node. We will complete this operation for a child node if there is a node in the subtree that needs a delivery, otherwise we do nothing. So, answer for current node will be

$$= \sum_{i = each\ child\ of\ current\ node} ans_i + 2$$

We can do this task with a single dfs.

Finally for each query of node u = dist[u]*2 + ans[u]; where, dist[u] = distance of node u from root and ans[u] = precalculated answer for node u.

Complexity: O(n + q) for precalculation dfs & queries.

Code: https://pastebin.com/EhdkyAxa

H. Save Stella!!

Author: Sanjida **_PEARL_** Nuri Pearl

It is a BFS shortest-path problem. At first, we should push all the bomb positions in a queue. Then run BFS and store the time of every cell when the cell will explode first. We don't need to calculate a cell's first explosion time twice because we are using BFS and it will find the minimum explosion time for each cell. Then again run another BFS for Stella and keep track of her time while running in that grid. If her time after entering a cell becomes greater than or equal to a cell's first explosion time, then she cannot enter that cell. She will have to run through other cells which have not exploded yet. If she

reaches the destination (n , m) cell before the 1st explosion time of this cell then she will be able to escape from the grid by the rescue helicopter!!!

Complexity : O(V + E)
Code: https://pastebin.com/QJsAqXsJ