
Analysis of task ezici

Solutions, analysis: Dimitar Shapatov; Idea, solutions, analysis: Kiril Zashev

Subtask 1

This subtask is longer, since initially it was a standalone one. In short, the statement asks us to split N agents into two groups so that in each group there is no distrust between two agents, that is, there is no edge between two of them for which $t = 0$.

In practice the task is purely an implementation one. In the author solution an approach with DFS is proposed, which successively visits vertices and gives them a "colour". This colour in fact tells us in which group a given agent has to be.

This means that when there is a connection between two vertices with $t = 1$, then these vertices have to be of the same colour, and if $t = 0$ they have to be of a different one.

If from a given vertex we can reach another one which is already coloured, then we do the following - we find what the new colour of the visited vertex has to be, and if it does not match the old one, this means that it is not possible to split the agents into two groups. Otherwise we continue the traversal, taking care not to run a DFS from an already visited vertex.

Solution with DSU (Mitko):

We can represent every agent as two vertices - one where he is in the first group and one where he is in the second one.

When two agents trust each other, we connect them in both groups. When two agents do not trust each other, we connect the first group of the first agent with the second group of the second agent and vice versa.

At the end, if the two groups of some agent are connected, this means that in order to satisfy the condition he has to be in both groups, therefore a contradiction arises.

Subtask 2

The task is to find the maximum depth of a tree.

Subtask 3

The task is to find the difference between the maximum and the minimum number in a sequence.

Subtask 4

The task is to find the number of pairs $i < j$ where $a_i + a_j$ is even. We can count the number of even numbers and the number of odd ones, after which our answer is $\frac{odd * (odd - 1)}{2} + \frac{even * (even - 1)}{2}$, since even+even=even, odd+odd=even.

Subtask 5

The task is to find the number of even numbers in a sequence.

Subtask 6

The task is to find the number of numbers which are strictly greater than all the ones before them.

Subtask 7

The task is to find the number of distinct numbers in the sequence.

Subtask 8

The task is to find some non-empty subsequence with a sum which is divisible by N without a remainder. This task is conceptually more complex compared to the previous ones. There is a proof that for every sequence there is a valid solution which is a contiguous subarray. If we look at the prefix sums modulo N , then we want to find two equal values. The proof is that the prefixes have $N + 1$ positions (we include the empty prefix), and only N valid values, therefore at least one value occurs at least twice.

Subtask 9

The task is to find the maximum sum of a contiguous subarray, which we can do by maintaining the current prefix and the minimum prefix sum up to this moment.



 Subtask 10

The task is to find the XOR of the sequence.