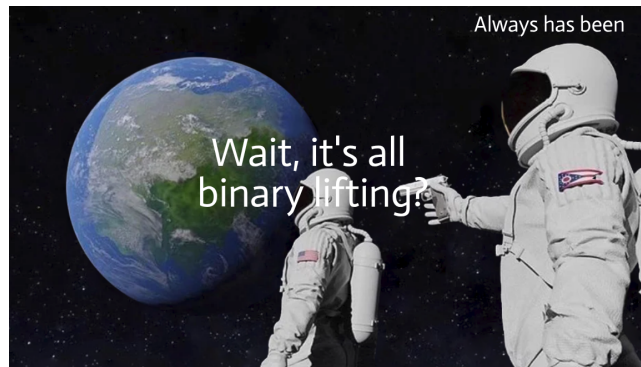


Analysis of task guess

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



At first glance the task is very easy – a standard binary search. What makes the task interesting are the forbidden operators – their goal is to prevent us from doing the standard binary search for this task. Besides that, the efficiency of our solution has to be the same as that of the binary search. How is this possible?

We suppose that quite a few of the participants have heard of binary lifting and LCA. In fact, here the solution for 100 points is very similar to finding the LCA of two vertices after we have levelled them to the same depth. The idea is the following – we try to “jump” by the largest power of two. If our query returns false, this means that we have jumped over the number we are looking for, therefore we try with the next smaller power of two. If the query returns true, we add the length of this jump to our answer and continue the search from the new position in the same way. In fact, what proves the solution of this task is that every binary search can be replaced by this binary jumping, but in the general case there is no point. The convenience here is that with this binary jumping the task can be written using only the operator + by computing powers of two with addition.

An interesting fact – since the operator / is forbidden in the task, we cannot use `#include <bits/stdc++.h>`.