

Analysis of task redica

Statement, solutions: Dimitar Shapatov; Analysis, solutions: Kiril Zashev

Subtask 0

This subtask is left for feedback from the system.

Subtask 1

With bruteforce we find all sequences which are valid and we try them.

Subtasks 2, 3

They have no intended solutions. They are left so that there are better partial points at smaller constraints, as well as to reward slower solutions.

Subtask 4

Here is the essential idea for solving the task. There is no way to avoid bruteforce for finding the correct sequence (if we want to get more than 10 points), but we have to choose a suitable suffix of the elements over which we will do the bruteforce, so that it behaves fast enough not to give TL. One indicator by which we can know approximately how slowly the bruteforce will behave is the number of valid sequences. We can find this number with dynamic programming.

Let $dp_{n,s}$ be the number of valid sequences which have n elements and a sum of the elements s . Then $dp_{n,s} = \sum_{i=1}^k dp_{n-1,s-i}$. With this we will already know the number of valid sequences that a given suffix of the sequence will have. Now it only remains to choose some number of possible sequences q at which the bruteforce will not behave too slowly, and to find the largest suffix with a number of ways $\leq q$. In the author solutions $q = 6 \cdot 10^6$ and $q = 8 \cdot 10^6$ are used.

Subtask 5

The slow part in the previous solution turns out to be the computation of the dp , because we compute every state with complexity $O(K)$. We can optimize the computation to $O(1)$ with prefix sums, since every state takes the sum of the dp values of some subarray of dp_{n-1} .