
Solutions

SUPER95

One way to solve the problem is to consider numbers of the form $x = A * 1000^B$, where A is an integer between 1 and 999 and B is an integer between 0 and 3.

All integers in the interval $[x, x+1000^B)$ start with the same letter $f(A, B)$. This allows us to process numbers in batches and significantly reduce the number of operations needed to find the solution.

KOKOS

Observe that the left and right halves of the graph are independent in that we can connect a vertex representing one of the K -th letters to any vertex representing one of the $(K+1)$ -th letters. Thus, the minimum number of vertices of the left half depends only on the first halves of all the words.

Let's add the half-words in order. The first half-word requires exactly K vertices. For each of the following half-words, we need to find the half-word already processed which is the most similar to it i.e. has the longest common prefix. If we sort the half-words lexicographically before starting, only pairs of adjacent words can share longest common prefixes.

If the length of this prefix is L , then $K-L$ new vertices must be added to the first half of the graph (the first L letters are already there, and we can't have cross-edges).

We proceed in a similar fashion for the right half of the graph, in fact we can reduce this case to the first one by reversing the second half-words and introduce edges in the opposite direction.

TROKUT

The brute-force solution uses $O(N^4)$ operations which is too many.

Consider two points A and B and the strip "below" the line segment connecting them i.e. the part of the plane bounded by three lines:

- 1) $x = x_A$
- 2) $x = x_B$
- 3) the line going through both points.

There are $O(N^2)$ such strips and we can calculate the number of other points inside the strip in $O(N)$ time. Then, for each possible choice of the three points that make a triangle, we can use the strips to calculate the number of points inside the triangle in $O(1)$ time. Thus, this solution is of complexity $O(N^3)$, both for the preprocessing part and finding the triangle.

BANANA

Given a number X (the candidate solution), we can efficiently determine if it is possible to achieve a maximum distance of D (where $D \leq X$) with K kunas. We can then use binary search to find the smallest such X .

For each vertex, we can recursively find the distance to the most distant vertex which is not closer to the starting vertex. Using these distances we can recursively check which distances can be improved by fixing roads.

Note that roads need to be fixed as close to the starting vertex as possible, because this opens way to improving distances from the starting vertex to many other vertices. Fixing any other road cannot yield a better solution.