Triangulation

Spoiler

Of course, we can only cut along diagonals which are part of the triangulation. For each one of them (name it d), we must check if it separates two triangles of the same color. Let's look at one of the colors, c. If we take the vertices of all triangles colored in c, $v_1^c < v_2^c < \cdots < v_{k(c)}^c$, then d can not cross any diagonal (v_i^c, v_j^c). But if it crosses some such diagonal, then it also crosses some diagonal (v_i^c, v_{i+1}^c) (where $v_{k(c)+1}^c = v_1^c$). When we realize that

$$\sum_{1 \le i \le n} k(i) \le 3n - 6$$

then we see that we have a solution running in $O(n^2)$.

Now we can add an data structure improvement. If we keep a two dimensional tree containing all diagonals (v_i^c, v_{i+1}^c), where each node of the first dimension tree contains a tree with ends of the diagonals which begin in a specified (for that particular node) segment. That kind of structure gives us chance to check if d crosses any of the diagonals in time $O(\log^2 n)$, which gives $O(n \log^2 n)$ in total.

There is another way to look at the problem, which gives even better results. We can observe that the graph, in which vertices represent triangles and there is an edge between vertices if the corresponding triangles share an edge, is a tree. Then the problem is to cut some of the edges of a tree not to separate vertices with the same color. Let's take the vertices of some color, find the lowest common ancestor of all these vertices and write the number of these vertices in the ancestor. Now, after doing that for each color, we have enough data to determine if an edge can be cut. More specifically: an edge cutting off a subtree can be cut, if the sum of the numbers written in all vertices of the subtree is equal to the number of vertices in the subtree. This leads to an algorithm with runtime $O(n \log n)$ (or even better, if we solve the LCA problem optimally).

Test data overview

no.	n	comment
1	96	Full binary tree depth 5, random colors
2	502	Several long paths, structured colors
3	2002	Many short paths, structured colors
4	5000	Medium length paths, structured colors
5	5000	Random tree, random colors
6	20002	Several long paths, structured colors
7	47978	Many short paths, structured colors
8	99979	Random tree, structured colors
9	100000	Many long paths, structured colors
10	98304	Full binary tree depth 15, random colors