

Tree Mirroring – spoiler

The problem can be split into two parts:

1. Find the leaves of the potential trees S and T (the vertices on the mirror plane).
2. Recreate the trees and verify that they are equal.

The first part is most difficult. To achieve that in $O(N)$ time, one can use the following algorithm:

- Find any cycle Q in G .
- If the edges of Q are removed from G , the connected components of the remaining graph are either single vertices or they contain exactly two of the vertices in Q , and these two are mirror images of each other in the tree-mirrored graph (if G is such graph). To find such a pair one can simply traverse this modified graph starting from any of the (non-isolated) vertices of Q until another vertex of Q is found.
- In the original graph, do separate breadth-first searches (BFS) starting from these two vertices. Any vertex that appears at the same distance from both starting vertices is a potential leaf.

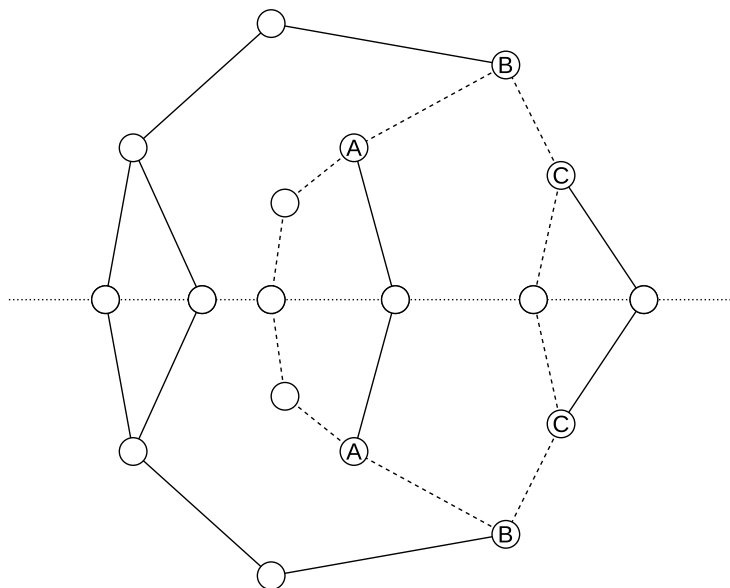


Figure 1: Example showing an arbitrary cycle in G (dashed line). Each of the three connected components of the modified graph (with the dashed edges removed) has two vertices in the cycle; these are labeled A for one component, B for another, and C for the third. Any of these pairs are mirror images of each other and thus have the same distance to all the leaves.

The second part can also be done in linear time. First, we divide the non-leaf vertices into the potential S and T trees by traversing the graph with the leaves removed (check for cycles that would make S



and T invalid trees). Now we construct a mapping between the vertices in S and T , for example by making a BFS starting from all leaves and identifying vertices that have the same corresponding child in the two trees. If no conflicts occur, the trees are identical.

There are also some special cases to keep track of. For example, if the graph G contains two vertices with degree 1 these should directly be used as mirror images (if the number of such vertices is not zero or two we can say NO directly). Moreover, if the graph is a single cycle, we just answer YES if and only if N is even.

For partial scores, the problem can also be solved in $O(N^2)$ time. For each vertex $v \in G$ with degree 2 we simply *assume* that v is a leaf node and thus its two adjacent vertices are mirror images of each other and can be used to find all the leaves as above. For each such potential set of leaves, we then perform the second part of the solution.