**Approach**:

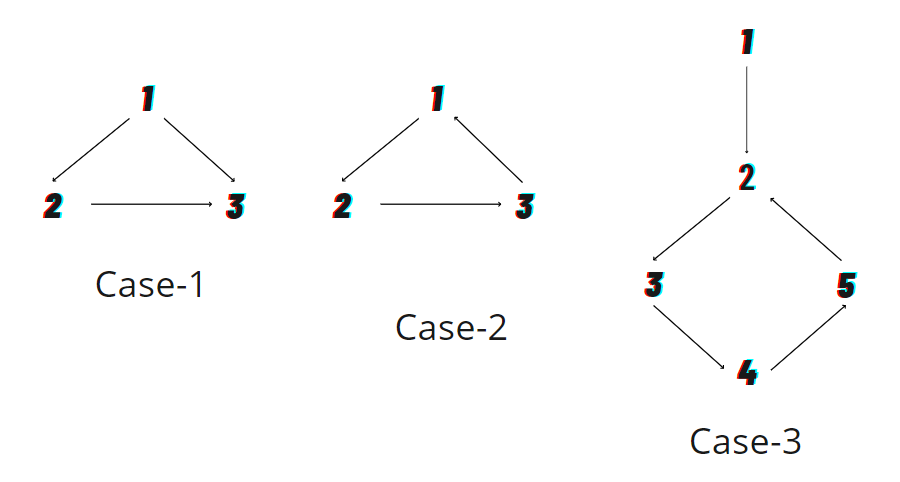
Firstly, I recommend solving "Redundant Connection" problem on Leetcode as the approach would be somewhat similar with slight additions to the code. The two questions are different in the sense that the "Redundant Connection" question deals with the detection of edges in an undirected graph which was pretty much a direct application of the DSU[Disjoint Set (Union-Find Algorithm)] algorithm. A cycle would have been surely present in case of a common parent between any two inter-connected nodes in the graph. But, in this question, if we try applying DSU directly, we would encounter an issue wherein two nodes would be having the same parent and would be connected but still won't be forming a cycle (See Diagram).

A diagram of a triangle with arrows and numbers

Description automatically generated

Now, In this problem, we would basically be facing three cases:

1. When a node has 2 parents
2. When an edge leads to a cycle
3. When there are both dual parents as well as a cycle in the graph.



We can find whether there are 2 parents of a node by counting the inwards-directed arrows for all nodes. If there is more than one inward arrow for any vertex/node, it means there is a dual-parent node. Hence, this states either case 1 or 3 is present. Then, we store both, the previous edge and the current edge in different variables and then compare both 1-by-1, removing one of the parent/edge and checking for a cycle using DSU.

Now the main question: Why can we apply DSU?

In the problem statement, it is clearly stated that the graph is initially a generic tree and a single additional edge is created/added in the tree. Hence, once we have checked and removed either of the dual parent edges (either of which must be the additional edge), we would no longer have a vertex having 2 parents and hence, cases like case-1 would not be there anymore. Now, we apply simple DSU, ignoring one dual parent edge at a time if it is present and then checking whether the graph is cyclic anymore.

Important Links:

1. Read about DSU: <https://www.scaler.com/topics/data-structures/disjoint-set/>