Problem 1192: Critical Connections in a Network

Problem Information

Difficulty: Hard
Acceptance Rate: 58.90%
Paid Only: No
Tags: Depth-First Search, Graph, Biconnected Component

Problem Description

There are `n` servers numbered from `0` to `n - 1` connected by undirected server-to-server `connections` forming a network where `connections[i] = [ai, bi]` represents a connection between servers `ai` and `bi`. Any server can reach other servers directly or indirectly through the network.

A _critical connection_ is a connection that, if removed, will make some servers unable to reach some other server.

Return all critical connections in the network in any order.

Example 1:

![](https://assets.leetcode.com/uploads/2019/09/03/1537_ex1_2.png)

Input: n = 4, connections = [[0,1],[1,2],[2,0],[1,3]] **Output:** [[1,3]] **Explanation:** [[3,1]] is also accepted.

Example 2:

Input: n = 2, connections = [[0,1]] **Output:** [[0,1]]

Constraints:

* `2 <= n <= 105` * `n - 1 <= connections.length <= 105` * `0 <= ai, bi <= n - 1` * `ai != bi` * There are no repeated connections.

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> criticalConnections(int n, vector<vector<int>>&
connections) {

}
};
```

Java:

```
class Solution {
public List<List<Integer>> criticalConnections(int n, List<List<Integer>>
connections) {

}
}
```

Python3:

```
class Solution:
def criticalConnections(self, n: int, connections: List[List[int]]) ->
List[List[int]]:
```