Assignment 1

As part of evaluation of this project everyone should maintain a github repo containing implementations of all algorithms. After creating repo, fill this form. Write well commented code and explain non trivial things in README.

Implement following as part of first assignment.

You will be given **undirected** graph as adjacency list. It contains n vertices numbered from 0 to n-1.

(Declare it as global variable or implement is as a class.)

1. DFS which visits all nodes and maintains tin and tout values for each node. Also implement a function which when given edge (u, v) classifies it into tree edge or back edge.
2. BFS which computes distance from a given vertex s. (For unreachable vertices put some value like INT_MAX or -1). Return distance vector.
3. Use DFS/BFS to return connected components of graph. If you have k connected components Give each connected component unique id from 1 to k. Return vector which maps each vertex to id of connected component it belongs to.
4. Use DFS/BFS to check if graph contains any cycle.
5. Modify bridge finding algorithm to delete all the bridges in graph.
    - You can not delete v from adjacency list of u in O(1) time. For deletion, just keep some marker to denote if the edge is deleted or not.
    - Graph is undirected. So deleting edge {u, v} means removing u from adjacency list of v as well as removing v from adjacency list of u.