WA8 - Assignment Group 16

Sravani Kirla - skirla
Matthew Heil - heilx069
Terrence Hanrahan - hanra028

1. DFS remains the same, only DFS-visit changes as shown below:

```
DFS-visit(G,u)
    time = time + 1
    u.d = time
    u.color = gray
    For each v in G.adj[u]
        If v.color == white
            print("Edge: ",u,",",v, " is a Tree edge")
            v.prev = u
            DFS-visit(G,v)
        If v.color == gray AND v.d < u.d
            print("Edge: ",u,",",v, " is a Back edge")
        If v.color == black AND v.prev != u AND u.d < v.d
            print("Edge: ",u,",",v, " is a Forward edge")
        If v.color == black AND v.d < u.d
            print("Edge: ",u,",",v, " is a Cross edge")
    u.color = black
    time = time + 1
    u.f = time

#notes. D is discover. F is finished
#Tree edges: u.d < v.d < v.f < u.f, v.prev = u, v must be white
#Back edge: v.d < u.d < u.f < v.f , v must be gray
#Forward edge: u.d < v.d < v.f < u.f and v.prev != u, v must be black
#Cross edge:v.d < v.f < u.d < u.f, v must be black
```

If G is undirected, only tree edges and back edges may exist. Thus, the forward edge and cross edge checks are not included in the modifications. Additionally, all edges are tree edges unless the edge from u to v goes to a gray or black vertex (excluding the vertices part of the edge).

2. If we use the textbook version of DFS and count the number of edges as we traverse the graph and store it in a temporary variable, if the number of edges are more than the number of vertices minus 1 i.e., |V|-1 then return true (indicating there is a cycle), if not then return false (indicating no cycle). This is because the minimum number of edges before a cycle is guaranteed in an undirected graph is (|V|-1). The algorithm will run in O(V) times as we exit method when we reach V vertices.

3.

WA8 - Assignment Group 16