CSE 417: Homework 2

Name: Qingchuan Hou

Student ID: 2127437

UWNetID: qhou

Problem 4:

The idea is using the search way similar to DFS, search every vertex's children untill find a children have a edges connecte to a vertex have pathed.

Pseudocode:

```
initial state: all vertices and edges undiscovered
initial a empty list V to store the path vertives

for i = 1 to n:                     // deal with unconnected graph
    if state v_i is undiscovered:
        mark v_i discovered
        Find(v_i):
            add v_i to path list V

            if have an undiscovered edges connect with v_i:
                mark edges (v_i, v_j) discovered

                if v_j is undiscovered:
                    mark v_j discovered
                    Find(v_j)
                else:
                    return the list V from v_j to v_i

            else:
                remove v_i from path list V
                Find(V[i-1])        // last previous element in V
```

Correctness Analysis:

This algorithm is similar to DFS, I only search one connected vertex v_j for the vertex v_i, then search the v_j's connected vertex. I also mark all the passed edges to discovered. In this case, when a vertex find a connected discoverd vertex through an undiscovered edges, it means we find a circle. To find the circle list, I store every passed vertex in a list, when I find the circle I pick up all the vertices after v_j. In this way, we may find some sub-tree is not a cycle but also stored in the list, so every time I find a vertex that is not connected to a circle, I remove it from the path list. Then go back to search the previous one from the list to see if it has a subtree that can make a circle. In this case, all the vertices remain from v_j to v_i will make a circle.

Run time

In this algrothm, for every connected component it only try to find one undiscovered connection every time. Each edges will be checked one time then remove by mark to discovered, the total number of edges is m. so the run time is $O(m)$. In the worst case, there may be n number of connected componoet in one graph, so the run time is $O(n)$. Thus, the runtime should be $O(n+m)$, same with DFS.