Gentry Atkinson
CS5329: Fall 2019
Homework 4

1)
Given G=(V,E) and Visited={}

```
Find_Cycle(Vertex  v, Vertex p ){
        If (n is in Visited) and (n is not p) return true
        Add n to Visited
        For c in n.neighbors{
                Find_Cycle(c, n)
        }
        Return false
}
```

Find_Cycle can be called on any vertex in V to start the search. A list is maintained of every visited vertex. Find_Cycle will be called at most V times (if there is no cycle) and is therefore O(V).

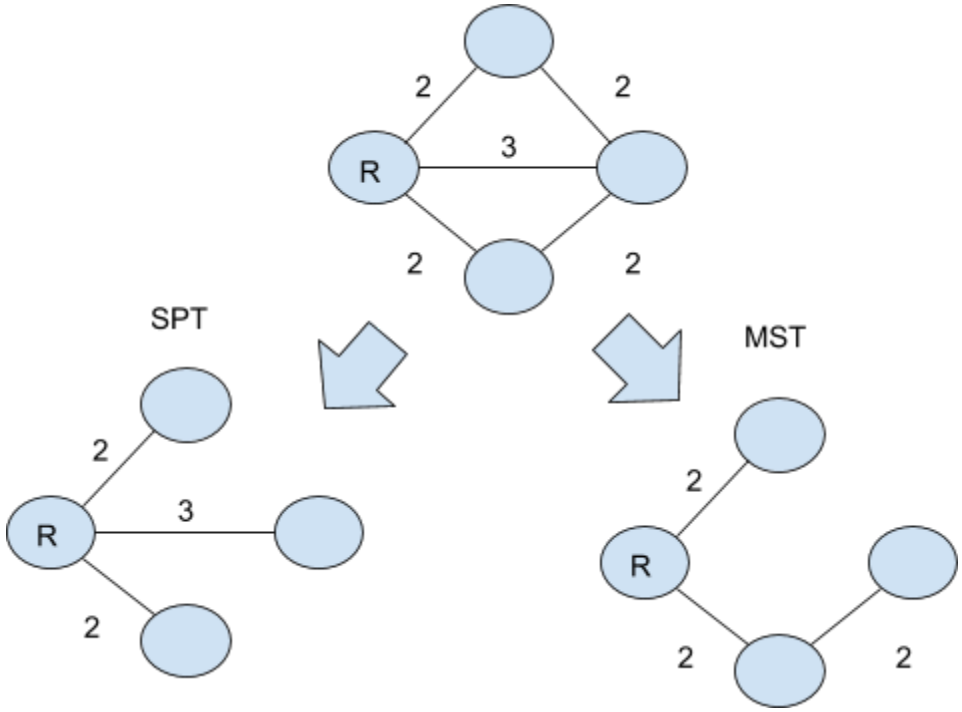
2)

a) DFS per Question 1
b) BFS because of the condition that the end node is near the start node. It will be better to check all nearby nodes first.
c) Either, since all vertices will be iterated over

3)

```
Find_shortest_BFS(Vertex start, Vertex finish){
        Queue next = {start}
        Distance = 0;
        While(next is not empty){
                V = next.pop()
                For (u in v.neighbors){
                        If (u = finish) return distance + 1
                        next.push(v)
                }
        }
        Return infinite
}
```

4) No



5)

Given G = (V,E)

```
Find_Max_Tree{
        Max_Tree = {};
        For e in E{
                Add e to Max_Tree;
                If G = (V, Max_Tree) has a cycle
                        Remove e from Max Tree
        }
        Return Max_Tee;
}
```

Check a graph for a cycle is O(V), so this algorithm has a runtime of O(V*E)

6)

	1	**2**	**3**	**4**
1	[1], 0	[1,4,2], 3	[1,4,2,3], 4	[1,4], 1
2	[2,3,1], 5	[2], 0	[2,3], 1	[2,3,1,4], 6
3	[3,1], 4	[3,1,4,2], 7	[3], 0	[3,1,4], 5
4	[4,2,3,1], 7	[4,2], 2	[4,2,3], 3	[4], 0

7)
The max flow from X to Y is 3