COP 5537 ASSIGNMENT #2

1 Pseudocode for Fleury's Algorithm

```
Algorithm 1 Fleury's_Algorithm (Adjacency matrix A, Vertex start, node_count)
 1: V \leftarrow \{v_1, v_2, ...., v_n\}

    ⊳ Set of vertices

 2: vertex_count ← node_count
                                                                                 ▶ Assign current number of vertices in graph A
 4: for v \leftarrow 0 to n do
        if A[start][v] then
                                                                                                 ▷ Checks if edge (start, v) exists
 5:
            visited\_nodes \leftarrow 0
                                                                                            ▶ Initialize all vertices as non-visited
 6:
 7:
            A[start][v] = A[v][start] = 0
                                                    ▶ Temporarily removing edge (start, v) to check if graph gets disconnected
 8:
            visited_count = Prims(A, n, v, visited_nodes)
                                                              ▶ Counting number of connected vertices using Prim's algorithm
 9:
            if abs(vertex_count - visited_count) \leq 1 then
                                                                         ⊳ Removing edge (start, v) will not separate A into two
10:
11:
                                                                           disconnected sets of edges
12:
                print edge (start, v)
13:
                if Bridge_edge(start, v) then
                                                                                                   ⊳ Edge (start,v) is bridge edge
14:
                    vertex_count = vertex_count - 1
                                                                                                    ⊳ Decrease vertex count by 1
15:
                if Bridge_edge(v, start) then
                                                                                                   ⊳ Edge (v,start) is bridge edge
16:
                    vertex_count = vertex_count - 1
                                                                                                    Decrease vertex count by 1 ⊳
17:
18:
                A[start][v] = A[v][start] = 0

⊳ Removing edge (start, v)

19:
                Fleury's_Algorithm (A, v, vertex_count)
20:
            else
21:
22:
                A[start][v] = A[v][start] = 1
                                                                                                   Not removing edge (start, v)
```

Algorithm 2 Prims(Adjacency matrix A, n, Vertex source, visited_nodes)

```
1: distance \leftarrow \infty
                                                                                                                      ▶ Initialization
 2: parent \leftarrow -1
 3: count \leftarrow 0
 4: distance[source] \leftarrow 0
                                                                                                       ▶ Set distance of source as 0
 5:
 6: for k \leftarrow 1 to n do
       u \leftarrow Min\_dis(distance, visited\_nodes)
                                                                   ▶ Returns node u with the shortest distance from source; if no
 7:
                                                                    connected node from source then returns -1
 8:
       if u \neq -1 then
                                                                                                9:
            visited\_nodes[u] = 1
10:
            count = count + 1
11:
12:
            for v \leftarrow 1 to n do
13:
                if visited_nodes[v] = 0 and A[u][v] > 0 and A[u][v] < distance[v] then
14:
15:
                    distance[v] = A[u][v]
16:
                    parent[v] = u
17: return count
```