

COP 5537 ASSIGNMENT #1

1 Pseudocode for Dijkstra's Algorithm

Algorithm 1 Dijkstra's Algorithm (Adjacency matrix A, Source s, Destination e)

```
1:  $V \leftarrow \{v_1, v_2, \dots, v_n\}$  ▷ Set of vertices
2:
3: for  $i \leftarrow 1$  to  $n$  do ▷ Initialization
4:    $\text{parent}[i] \leftarrow -1$  ▷ Parent of  $V_i$  not assigned yet
5:    $\text{color}[i] \leftarrow 0$  ▷ Vertex  $V_i$  not visited yet
6:    $\text{distance}[i] \leftarrow \infty$  ▷ Distance of  $V_i$  from source node not calculated yet
7:  $\text{distance}[s] \leftarrow 0$  ▷ Starting to visit the graph from source node
8:
9: for  $i \leftarrow 1$  to  $n$  do ▷ Repeat for all the vertices
10:   $v \leftarrow \text{Min\_dis\_vertex}(n)$  ▷ Returns the vertex currently having the minimum distance from the source
11:   $\text{color}[v] \leftarrow 1$  ▷ Mark vertex v as visited
12:
13:  for all vertices u adjacent to v do
14:    if  $\text{distance}[u] > \text{distance}[v] + A(u,v)$  and  $\text{color}[u] = 0$  then ▷  $A(u,v)$  is the edge-weight between vertex u and v
15:       $\text{distance}[u] \leftarrow \text{distance}[v] + A(u,v)$  ▷ Update distance of vertex u
16:       $\text{parent}[u] \leftarrow v$  ▷ Assign v as u's parent
17:
18:  $\text{Print\_shortest\_path}(e)$  ▷ Outputs the shortest path between the starting vertex s and ending vertex e
19: return  $\text{distance}[e]$  ▷ Returns the shortest path between the starting vertex s and ending vertex e
```

Algorithm 2 Min_dis_vertex(n)

```
1:  $\text{min\_val} \leftarrow \infty$ 
2: for  $i \leftarrow 1$  to  $n$  do
3:   if  $\text{color}[i] = 0$  and  $\text{distance}[i] < \text{min\_val}$  then
4:      $\text{index} \leftarrow i$ 
5: return index ▷ Returns the index of the non-visited vertex having minimum distance from source node
```

Algorithm 3 Print_shortest_path(v)

```
1: if  $\text{parent}[v] = -1$  then
2:   return
3:  $\text{Print\_shortest\_path}(\text{parent}[v])$  ▷ Recursion
4: Output v
```
