FALL 2019 – CS 303 Algorithms and Data Structures
Lab 12
Deadline: November 17, 2019 Sunday 11:59pm

Notes:

• Implement the algorithm and analyze the results using the give input files
• Deliverables: Report.pdf file and your code file (please do not send a zip file. If you have more than one class in your code, then submit each file separately through Canvas.)
• Homework report must follow the guidelines provided in the sample report uploaded in Canvas

Objectives:

- Implement Depth First Search algorithm on graphs.
- Implement Topological Sort

Problems:

1. Write a program to implement the depth-first search algorithm using the pseudocode given on the next page.

2. Write a driver program, which reads input file mediumG.txt as an undirected graph and runs the depth-first search algorithm to find paths to all the other vertices considering 0 as the source. This driver program should display the paths in the following manner:

   0 to 'v': list of all the vertices traversed to go to v from 0, separated by ',' . There is another file called tinyG.txt which you can use for your testing purpose.

3. Implement the Topological Sort algorithm for directed graphs and run the algorithm on the directed graph that is provided in the tinyDG.txt file. The pseudocode is provided in the next page.

```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)
```

```
DFS-VISIT(G, u)
 1  time = time + 1              // white vertex u has just been discovered
 2  u.d = time
 3  u.color = GRAY
 4  for each v ∈ G.Adj[u]        // explore edge (u, v)
 5      if v.color == WHITE
 6          v.π = u
 7          DFS-VISIT(G, v)
 8  u.color = BLACK              // blacken u; it is finished
 9  time = time + 1
10  u.f = time
```

TOPOLOGICAL-SORT(*G*)

1 call DFS(*G*) to compute finishing times *v.f* for each vertex *v*
2 as each vertex is finished, insert it onto the front of a linked list
3 **return** the linked list of vertices