

Fall 2023 – CS 303 Algorithms and Data Structures  
Lab 09

**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  $u \in G.V$ 
2       $u.color = \text{WHITE}$ 
3       $u.\pi = \text{NIL}$ 
4   $time = 0$ 
5  for each vertex  $u \in G.V$ 
6      if  $u.color == \text{WHITE}$ 
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   $u.color = \text{GRAY}$ 
4  for each  $v \in G.Adj[u]$       // explore edge  $(u, v)$ 
5      if  $v.color == \text{WHITE}$ 
6           $v.\pi = u$ 
7          DFS-VISIT( $G, v$ )
8   $u.color = \text{BLACK}$          // 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
```

Fall 2023 – CS 303 Algorithms and Data Structures  
Lab 09

**DATA**

largeG.txt

medium.txt

tinyDG.txt

**Grading Rubric**

Coding	Implementing Algorithms	20 points
	Producing Correct Outputs	20 points
Report	Explaining the algorithms used	10 points
	Displaying the output with a graph or table	20 points
	Comparing the outputs and discussing the time complexity of algorithms	20 points
	Correct submissions of the files (ICF, Code.zip, report.pdf)	10 points