

## Programming assignment 8.

**Due date:** Friday, May 3, 2019 at 11:00pm

.....

In this program you are required to implement the BFS and DFS algorithms.

1. Request the user to determine the order ( $|V|$ ) and size ( $|E|$ ) of the graph.
2. Generate  $|E|$  random edges into the adjacency matrix/list (**Adj**) to make a random *directed* graph.
3. Print the resulting adjacency matrix/list.

### Part A.

1. Request the user to determine the starting vertex ( $u$ ) for **BFS** and **DFS\_visit** algorithms
2. Call *BFS* function to find the vertices reachable from vertex  $u$  and print the *shortest paths* and their *lengths/distances*.
3. Call *DFS\_visit* function to find the vertices reachable from vertex  $u$  and for each vertex print the *start/finish time*.

### Part B. In this part, we print the topological order of the vertices

1. Run *DFS* function to check if the graph is a DAG (directed acyclic graph):
  - ✓ Search for backward edges. If there are any, (the graph has a cycle.) print: "Cycle detected, topological sort is impossible".
2. If the graph is DAG, (while running DFS):
  - ✓ Insert the vertex into a linked list as it finishes.
  - ✓ Using your linked list, print the topological order of the vertices along with their *start/finish time*.