Omar Khattab - 300202727

Université d'Ottawa
Faculté de génie

University of Ottawa
Faculty of Engineering

uOttawa
L'Université canadienne
Canada's university

École de science informatique et de génie électrique (SIGE)

School of Electrical Engineering and Computer Science (EECS)

Assignment 10 Graph Traversals (3% - 12 points)

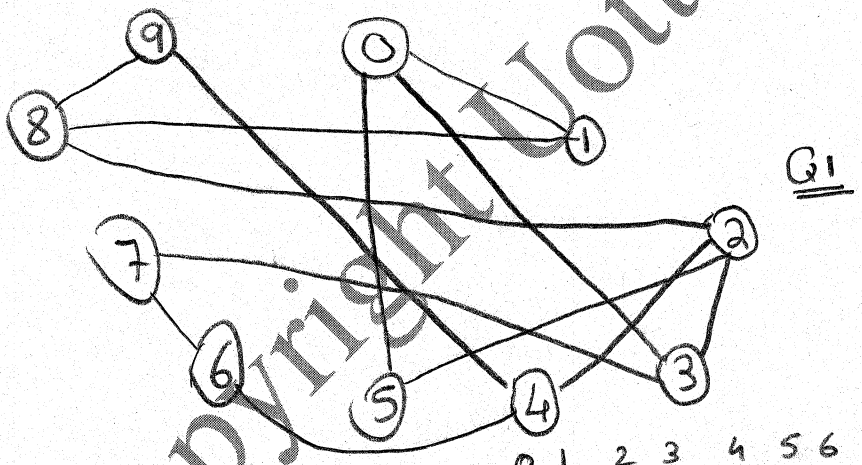
CSI2110 (Fall 2021)

Due date: Thursday November 25

Late assignment policy: ***1min-24hs late are accepted with 30%off; no assignments accepted after 24hs late.***

Below is an undirected graph represented using adjacency lists:

0: (0,1), (0,5), (0,3)
1: (1,0), (1,8)
2: (2,5), (2,3), (2,4), (2,8)
3: (3,0), (3,2), (3,7)
4: (4,2), (4,9), (4,6)
5: (5,0), (5,2)
6: (6,7), (6,4)
7: (7,3), (7,6)
8: (8,1), (8,2), (8,9)
9: (9,8), (9,4)



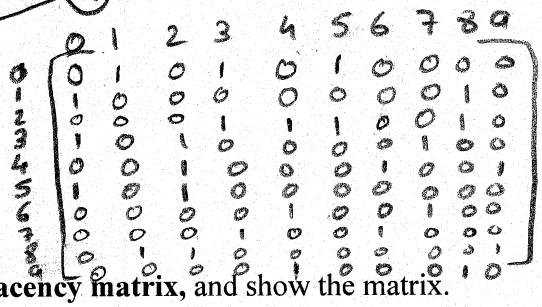
Q1

Question 1. **[2 points]**

Draw the graph corresponding to the given adjacency lists.

Question 2. **[2 points]**

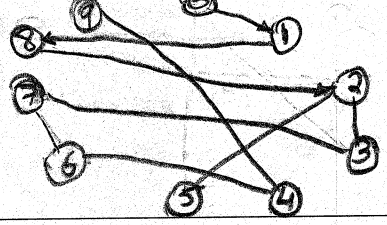
Change the representation of the graph from adjacency lists to **adjacency matrix,** and show the matrix.



Question 3. **[4 points = 3+1]**

3a) Using the DFS algorithm in the Appendix, perform a depth-first search traversal on the given graph starting from node **0** and using the **adjacency lists** representation of the graph. The adjancency lists will influence the order in which the vertices are considered; for example G.incidentEdges(0) will return the list : (0,1), (0,5), (0,3) so that the edges will be considered in this order.

List the vertices in the order they are visited, and list the edges in the order they are labeled by the algorithm, displaying their labels.



Vertices in order of visit: 0, 1, 8, 2, 5, 3, 7, 6, 4, 9

Edges and labels in order of visit: (0,1) D, (1,0) B, (1,8) D, (8,1) B, (8,2) D, (2,5) D, (5,0) B, (5,2) B, (2,3) D, (3,0) B, (3,2) B, (3,7) D, (7,3) B, (7,6) D, (6,7) B, (6,4) D, (4,2) B, (4,9) D, (4,6) B, (2,4) B, (2,8) B, (8,9) B, (0,5) B, (0,3) B

D → Discovery
B → Back

Please give the edges in the order they are labeled, display each edge in the direction of visit, and use the first letter of the label: Example - if a discovery edge was found coming from vertex b to a, the entry for this edge will be displayed "(b,a) D, "

3b) Suppose DFS is run starting at vertex 0 using the same adjacency lists, but instead of the recursive algorithm you use the algorithm that uses a Stack. The edges incident to a vertex v are given by the same method G.incidenceEdges(v), and when inserted in the stack this order is followed.

Does the order in which vertices are visited change? If so, give the vertices in order of visit; if not explain why not.

Order does not change because stacks are first in last out which acts and give same result as the recursive approach.

Question 4. **[4 points = 3+1]**

4a) Using the BFS algorithm in the Appendix, perform a breadth-first search traversal of the graph starting from node **0** and using the **adjacency lists** representation of the graph.

List the vertices in the order they are visited circling the groups of vertices that belong to each list L_0, L_1, L_2, etc. List the edges in the order they are labeled by the algorithm, displaying their labels.

Please, use similar format as suggested in question 3a.

L0 L1 L1 L1 L2 L2 L2 L3 L3 L3

Vertices in order of visit: 0, 1, 5, 3, 8, 2, 7, 9, 4, 6

Edges and labels in order of visit: (0,1) D, (0,5) D, (0,3) D, (1,0) C, (1,8) D, (5,0) C, (5,2) D, (3,0) C, (3,2) C, (3,7) D, (8,1) C, (8,2) C, (8,9) D, (2,5) C, (2,3) C, (2,4) D, (2,8) C, (7,3) C, (7,6) D, (9,8) C, (9,4) C, (4,2) C, (4,9) D, (4,6) C, (6,7) C, (6,4) C

D → Discovery
C → Cross

4b) Suppose BFS is run starting at vertex 0 using the same adjacency lists, but instead of the algorithm used in part a, you use the algorithm that uses a single Queue to process the vertices. The edges incident to a vertex v are given by the same method G.incidenceEdges(v), and when inserted in the queue this order is followed.

Does the order in which vertices are visited change? If so, give the vertices in order of visit; if not explain why not.

Order does not change because the algorithm loops on the edges in the order they are in the adjacency list while the queue will act the same as it is first in first out, meaning it will also be in order of the adjacency list