**CMP4272- DATA STRUCTURES AND ALGORITHMS**

**LAB SESSION-5: Graphs**

**Objectives:**

* Use graphs and networks to model problems in computing.
* Identify standard graph/network problems and algorithms relevant to computing applications.
* Understand the *edge list, adjacency list*and*adjacency matrix* data structures for graphs and networks.

1. **Background:**

**Graph:** A data structure that consists of a set of nodes (*vertices*) and a set of edges that relate the nodes to each other. The set of edges describes relationships among the vertices.

There are 3 components which define a graph data structure:

1. Edge lists
2. Adjacency lists
3. Adjacency matrices

For the graph given below:

A diagram of a network

Description automatically generated

* The graph is an undirected graph having 5 vertices and 7 edges.
* Vertices: (0,1,2,3,4)
* Edges: (0,1) , (0,2), (1,2), (1,3),(1,4), (2,4), (3,4)

A table with numbers and a number in it

Description automatically generated

A table with numbers and text

Description automatically generatedA table with numbers and letters

Description automatically generated

1. **Complete the following exercises before you attempt the mandatory ones provided under Lab submission Exercises in the next section.**

**Exercise-1**: The following Python code represents a graph using the edge list data structure.

* 1. Draw a node/link diagram for the graph.

A group of letters and symbols

Description automatically generated

*Answer:*

* 1. *Two node or vertices are adjacent if they are connected to each other through an edge*. List adjacent vertices of each of the following vertices:
     + vertex **C**
     + vertex **F**

*Answer:*

**Exercise-2:** In an adjacency list representation, each vertex in the graph (*directed*) is associated with a list of its neighbouring vertices. The vertex can be represented in Python as:

class Node:

def \_\_init\_\_(self, value):

self.value = value

self.neighbours = []

1. Write necessary Python code to create 4 vertices with value A, B, C and D.

*Answer:*

1. Draw a node/link diagram of the vertices (nodes) that you created above.

*Answer:*

1. Refer to the diagram created above (in ii); can it be regarded as a graph? Explain.

*Answer:*

1. Consider the Python code given below, the function add\_neighbour adds neighbours to a vertex.

class Node:

def \_\_init\_\_(self, value):

self.value = value

self.neighbours = []

def add\_neighbour(self, node):

self.neighbours.append(node)

Using add\_neighbour function, write necessary Python code to add the neighbours for each node as given in the adjacency:

|  |  |  |  |
| --- | --- | --- | --- |
| **Adjacency List** | | | |
| **A:** | B | C |  |
| **B:** | C |  |  |
| **C:** | D |  |  |
| **D:** | A | B |  |

*Answer:*

1. Draw updated node/link diagram of the graph.

*Answer:*

1. **Lab Submission Exercises:**

* **Submitting the solution of the following exercises (3 and 4) is mandatory.**
* **Solutions that comprise of python code, must be well documented. (Include necessary comments)**

**Exercise-3:** Consider the following graph and the Python code:

A diagram of a network

Description automatically generated

class Node:

def \_\_init\_\_(self, value):

self.value = value

self.neighbours = []

def add\_neighbour(self, node):

self.neighbours.append(node)

1. Write necessary Python code that creates the vertices and edges of the graph.

**A screen shot of a computer program

Description automatically generated**

1. Draw the adjacency matrix of the graph.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Vertex 0** | **Vertex 1** | **Vertex 2** | **Vertex 3** | **Vertex 4** |
| **Vertex 0** | 0 | 1 | 1 | 0 | 0 |
| **Vertex 1** | 1 | 0 | 1 | 1 | 0 |
| **Vertex 2** | 1 | 1 | 0 | 0 | 1 |
| **Vertex 3** | 0 | 1 | 0 | 0 | 1 |
| **Vertex 4** | 0 | 0 | 1 | 1 | 0 |

**Exercise-4:** Consider the following graph.

A diagram of a network

Description automatically generated

Traversal algorithms visit each of the vertices of the graph.

A screenshot of a computer program

Description automatically generated

Show the necessary steps of the following traversals, with starting node as 1.

1. Breadth First Search (BFS)

|  |  |
| --- | --- |
| Node Visited | Queue |
| 1 | 1 |
| 2 | 1 2 |
| 3 | 1 2 3 |
| 4 | 1 2 3 4 |
| 5 | 1 2 3 4 5 |

A diagram of a network

Description automatically generated with medium confidence

1. Depth First Search (DFS)

|  |  |
| --- | --- |
| Node Visited | Stack |
| 1 | 1 |
| 2 | 1 2 |
| 3 | 1 2 3 |
| 4 | 1 2 3 4 |
| 5 | 1 2 3 4 5 |

A diagram of a triangle with lines and dots

Description automatically generated

1. **Moodle Submission:**

You are required to submit your solution in the word document.

Naming Format: **StudetName\_studentID.docx [ or other word formats]**

Example : AliceSmith\_514099.docx

**NOTE**

* It is important to complete the weekly labs in particular labs 2, 3, 4, 5 and 6 because it contains questions that are part of the coursework. (Weightage: 25%).
* Only one of these labs will be chosen randomly for marking, so it is important that you complete and submit each of these labs.
* Ideally, you should submit each lab within one week of that lab session.
* Solutions that comprise of python code, must be well documented. (Include necessary comments)