DATA STRUCTURES
FALL 2021

LAB 15

University of C

Learning Outcomes

In this laboratory, you will implement the Graph ADT using Adjacency List.

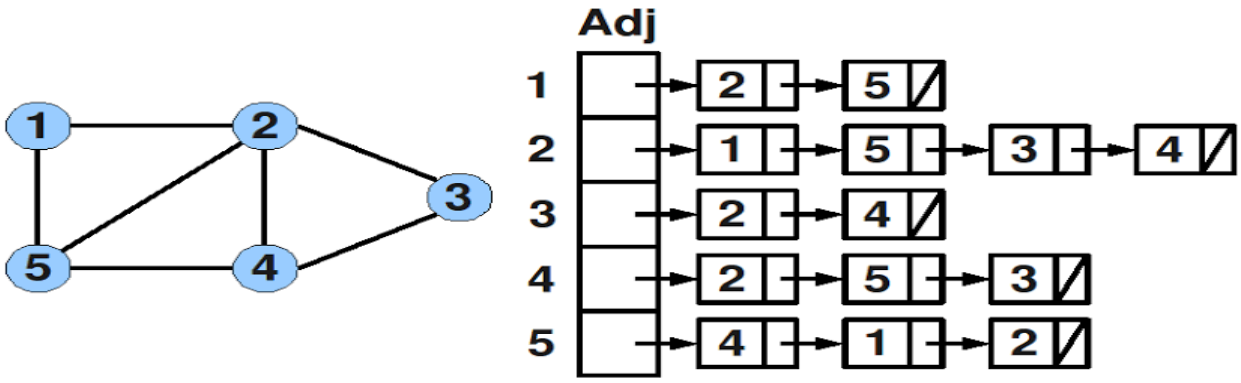
Graph ADT

Data Items

Each vertex in a graph has a label (of type int) that uniquely identifies it. Vertices may include additional data.

Structure

The relationship between the vertices in a graph is expressed using a set of directed/undirected Edges, where each edge connects one pair of vertices. Here is an example of an undirected graph and its representation in the form of adjacency list.



Task 1

Create class Graph and implement the following functions:

Graph (int maxVertices)

Requirements:
None

Results:
Constructor. Creates an empty graph. Allocates enough memory for a graph containing maxNumber vertices.

void addEdge (int src, int dest)

Requirements:
Graph includes vertices src and dest.

Results:
Inserts an undirected edge connecting vertices src and dest into a graph.

void showGraphStructure ()

Requirements:
None

Results:
Outputs a graph with the vertices in array form and the edges in adjacency list form. If the graph is empty, outputs “Empty graph”. Note that this operation is intended for testing/debugging purposes only.

Task 2

Using the Queue data structure you have created in the previous labs, implement Breadth First Traversal function.

void BFT (int startNode)

Requirements:
Graph contains the start node. You can create an array of visited nodes (size is same as size of graph).

Results:
Outputs nodes of the graph in BFT.

National University of Computer & Emerging Sciences (
School o