UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

Submitted by:
Mamanao, Kurt Marwin, C.

Instructor:
Engr. Maria Rizette H. Sayo

OCT, 11, 2025

I. Objectives

Introduction

A graph is a visual representation of a collection of things where some object pairs are linked together. Vertices are the points used to depict the interconnected items, while edges are the connections between them. In this course, we go into great detail on the many words and functions related to graphs.

An undirected graph, or simply a graph, is a set of points with lines connecting some of the points. The points are called nodes or vertices, and the lines are called edges.

A graph can be easily presented using the python dictionary data types. We represent the vertices as the keys of the dictionary and the connection between the vertices also called edges as the values in the dictionary.

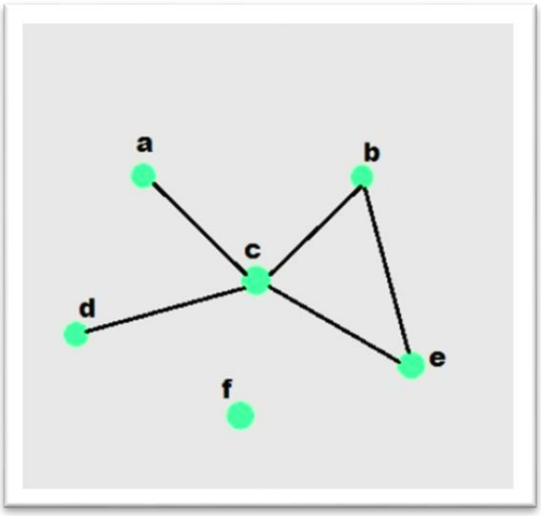

Figure 1. Sample graph with vertices and edges

This laboratory activity aims to implement the principles and techniques in:

- To introduce the Non-linear data structure – Graphs
- To discuss the importance of Graphs in programming

II. Methods

A. Discuss the following terms related to graphs:

1. Undirected graph
2. Directed graph
3. Nodes
4. Vertex
5. Degree
6. Indegree
7. Outdegree
8. Path
9. Cycle
10. Simple Cycle

III. Results

A. Discussion of Graph Terminologies

1. **Undirected Graph**
   An undirected graph is a type of graph where the edges have no direction. The connections between vertices go both ways, meaning if there is an edge from A to B, there is also an edge from B to A.
2. **Directed Graph**
   A directed graph (or digraph) is a graph where each edge has a specific direction. It connects one vertex to another in only one way.
3. **Nodes**
   Nodes are the basic units or points in a graph that represent entities, objects, or data. They are also called vertices.
4. **Vertex**
   A vertex is a single point in the graph where edges meet or connect. It is another term for node.
5. **Degree**
   The degree of a vertex is the total number of edges connected to it.
6. **Indegree**
   Indegree refers to the number of edges directed *towards* a vertex in a directed graph.
7. **Outdegree**
   Outdegree refers to the number of edges that *leave* a vertex in a directed graph.
8. **Path**
   A path is a sequence of vertices connected by edges. It shows a way to travel from one vertex to another.
9. **Cycle**
   A cycle is a path where the starting and ending vertex are the same, and no edge is repeated.
10. **Simple Cycle**
    A simple cycle is a cycle that does not visit any vertex more than once, except for the starting/ending vertex.

IV. Conclusion

In this activity, I was able to understand more about graphs and their parts like vertices, edges, path, cycle, and other basic terms. I'm still new to this topic, but it helped me remember some of the things we learned before in class. I learned the difference between directed and undirected graphs and how nodes are connected.

Even if it’s a bit confusing at first, this lesson gave me a better idea of how graphs work in programming. I know I still need more practice, but this is a good start for me to understand it better.

References

[1] GeeksforGeeks, “Graph Data Structure and Algorithms,” *GeeksforGeeks*, 2024. [Online]. Available: https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/