



UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

Submitted by:
Asugas, Kenneth R.

Instructor:
Engr. Maria Rizette H. Sayo

10, 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.

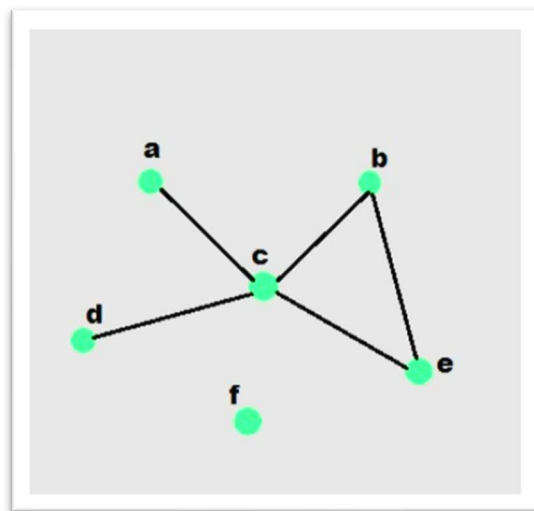


Figure 1. Sample graph with vertices and edges

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

1. Undirected graph

This figure represents an undirected graph because the lines (edges) connecting the points (vertices) do not have arrows, meaning the connections have no specific direction.

2. Directed graph

This graph is not a directed graph. A directed graph would have arrows on the edges to show one-way connections.

3. Nodes

The nodes (or vertices) in the graph are the labeled points: a, b, c, d, e, and f.

4. Vertex

The points such as a, b or, are called vertices.

5. Degree

The degree is the number of edges connected to a vertex.

Degree(c): 4 (edges to a, b, d, e)

Degree(b): 2 (edges to c, e)

Degree(e): 2 (edges to c, b)

Degree(a): 1 (edge to c)

Degree(d): 1 (edge to c)

Degree(f): 0 (It is an isolated vertex with no edges)

6. Indegree

Indegree is only applicable to directed graphs. Since this is an undirected graph, the term is not directly applied here.

7. Outdegree

Outdegree is only applicable to directed graphs. Since this is an undirected graph, the term is not directly applied here.

8. Path

A path is a sequence of connected vertices. Example Path: $a > c > d$

9. Cycle

A cycle is a path that starts and ends at the same vertex. Example Cycle: $b > c > e > b$

10. Simple Cycle

A simple cycle is one that does not revisit any intermediate vertex.

Example Simple Cycle: $b > c > e > b$ is also a simple cycle, as vertices are visited only once.

IV. Conclusion

This lab report is dedicated to the Intro to Graphs, which is a talk about non-linear data structures. A graph is defined as a visual form of a set of items in which some pairs of objects are connected together. An undirected graph, or simply a graph, is made up of a set of points joined by lines. The vertices are called points or nodes, and the connecting lines are referred to as edges. The most important terms used in graphs are the Degree of a vertex (the number of edges attached to it), and in the case of directed graphs, the Indegree and Outdegree (edges entering or leaving a vertex, respectively). Connectivity is defined in terms of a Path (a connected sequence of vertices), and a Cycle is a path which starts and ends at the same vertex. A Simple Cycle is a cycle which does not return to any of the intermediate vertices. The significance of graphs in programming is elaborated upon, and in Python, graphs can be depicted using dictionary data types, with vertices being keys and their linkages (edges) as values.

References

[1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.