UNIVERSITY OF CALOOCAN CITY
COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

Submitted by:
Uy, Junichiro H.

Instructor:
Engr. Maria Rizette H. Sayo

October 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.

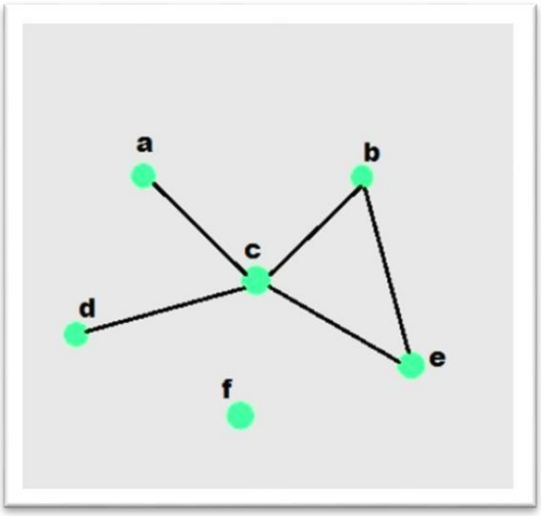

Figure 1. Sample graph with vertices and edges

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:

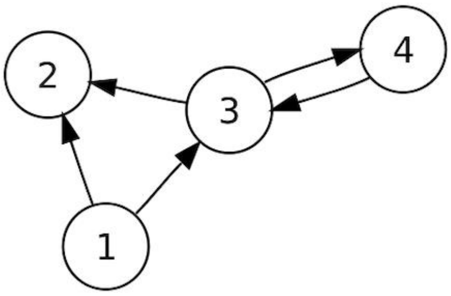

Figure 1 Picture of graph from study.com

- This graph doesn't have a fixed direction. You can go back to the same node before, and vice versa. This is a two-way street unlike directed graphs. This also doesn't have arrows.

2. Directed Graph:

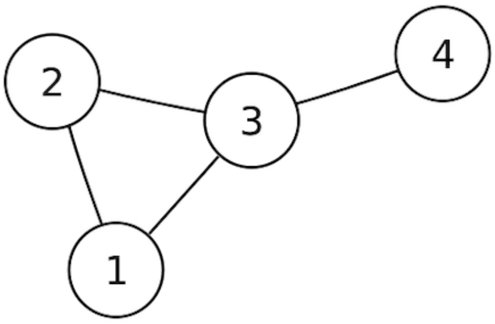

Figure 2 Picture of graph from study.com

- The graphs have arrows pointing to certain nodes. This means that it has a fix direction and you can't go back to the same node you left. It is a one-way street unlike Undirected Graphs.

3. Nodes: These are the points, and the fundamental elements of the graph. It is usually represented as a point or a circle, with an alphabet or number.
4. Vertex: This is just the same as a node, it is a point in the graph.
5. Degree: Degree is the highest number of connections a node has in a graph. Let's say that there are 4 nodes connected to node c, then that means the degree of the graph is 4.
6. Indegree: This is about a directed graph, it is the number of nodes pointing towards a single node. Let's say the there are 4 nodes pointing to node c, so the indegree would be 4.
7. Outdegree: This is the opposite of indegree, it is basically the number of nodes pointing outward or pointing away at a node. Let's say there are 4 nodes pointing away at node c, so the outdegree would be 4.
8. Path: This is simply the sequence of nodes or vertices connected by edges. Let's say you can connect node a to node b to node c to node d, it would be called a path.
9. Cycle: This is just a sequence that starts and ends with the same node. Let's say node a to node b to node c then to node a again.

10. Simple Cycle: This is a sample of cycle, but it is strict. A simple cycle can start and end with the same node, but it can’t repeat nodes inside it. Simple cycle: a, b, c, d, a. Un simple cycle: a, b, c, d, c, a.

IV. Conclusion

In conclusion, every terminology used in this laboratory is helpful in navigating graphs of data. This is going to be an invaluable and an important introductory resource that will help us in understanding data graphs of the company we will be working in the future. This will give us a kickstart on solving complex data connections.

References

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

[2] https://study.com/academy/lesson/weighted-graphs-implementation-dijkstra-algorithm.html