Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Ramos, Jan Lawrence M. | Engr. Maria Rizette H. Sayo |

October 11, 2025

# 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.

A diagram of a triangle with green dots

AI-generated content may be incorrect.

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

# Methods

* 1. 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

# Results

**A. Discussion of Graph-Related Terms**

1. **Undirected Graph:**

* A graph in which the edges have no direction. The relationship between vertices is bidirectional. If there is an edge between vertex A and vertex B, you can traverse from A to B and from B to A.

1. **Directed Graph (Digraph):**

* A graph in which the edges have a direction. The relationship is one-way, indicated by an arrow. An edge from A to B does not imply an edge from B to A.

1. **Nodes/Vertices:**

* The fundamental units of a graph. They represent the entities or objects in the network (e.g., people in a social network, cities on a map).

1. **Edge:**

* A connection between two vertices. It represents the relationship between the entities.

1. **Degree:**

* In an undirected graph, the degree of a vertex is the number of edges incident to it.

1. **Indegree:**

* In a directed graph, the indegree of a vertex is the number of edges coming into it.

1. **Outdegree:**

* In a directed graph, the outdegree of a vertex is the number of edges going out of it.

1. **Path:**

* A sequence of vertices where each adjacent pair is connected by an edge. For example, A-B-C-D is a path from A to D.

1. **Cycle:**

* A path that starts and ends at the same vertex, with no repeated edges.

1. **Simple Cycle:**

* A cycle that does not repeat any vertices (except the start and end vertex).

# Conclusion

In conclusion, this laboratory activity successfully helped us understand the basics of graphs by learning the key terms and seeing how a graph can be built in Python using a simple dictionary. We learned that graphs are just collections of points (vertices) connected by lines (edges), and this simple structure is powerful for modeling real-world connections like social networks or maps, making them a very important tool in programming.

**References**

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

[2] "Graph (discrete mathematics)," Wikipedia. [Online]. Available: <https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)>. [Accessed: Oct. 11, 2023].

[3] M. Newman, *Networks*. Oxford, UK: Oxford University Press, 2018.