Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Tan, Charles Dominic S. | 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

1. Undirected Graph - An undirected graph is a collection of spots connected by lines that allow for two-way movement.

Examples:

V = {(A, C), (C, A), (B, C), (C, B), (B, E), (E, B), (C, D), (D, C), (C, E), (E, C)}

2. Directed Graph - A way of visualizing a group of points is via directed graphs, in which arrows point from one vertex to another.

Examples:

V = {A → C}

V = {C → A}

V = {B → C}

V = {C → B}

V = {B → E}

V = {E → B}

V = {C → D}

V = {D → C}

V = {C → E}

V = {E → C}

3. Node - A node is a single point in a graph that can connect with other points.

Examples:

V = {A, B, C, D, E, F}

4. Vertex - Is a basic unit used to represent a specific point in a graph.It is a crucial part of

data storage and has edges connecting it to other vertices.

Examples:

V = {A, B, C, D, E, F}

5. Degree - The degree of a node is the number of lines or arrows connected to it.

Example:

A = 1

B = 2

C = 4

D = 1

E = 2

F = 0

6. Indegree - Indegree is the number of arrows that point **to** a node.

Examples:

A = 1

B = 0

C = 2

D = 1

E = 2

F = 0

7. Outdegree - Outdegree is the number of arrows that go **out from** a node.

Examples:

A = 1

B = 2

C = 2

D = 0

E = 0

F = 0

8. Path - A path is a way to go from one node to another by following the arrows or lines.

Examples:

A → D

V = {(A, C), (C, D)

D → E

V = {(D, C), (C, E)}

A → E

V = {(A. C), (C, E),

V = {(A, C), (C, B), (B, E)}

B → D

V = {(B, C), (C, D)}

E → A

V = {(E, B), (B, C), (C, E)}

9. Cycle - A cycle is a path that starts and ends at the same node without using the same line twice.

10. Simple Cycle - A simple cycle is a loop where you don’t visit the same node twice, except for the first and last one.

# Conclusion

In this exercise, we focused on exploring the concept of graphs and their application in depicting the relationships among the points. As a part of this, we implemented a graph in Python, which consisted of vertices and connections between them. Additionally, we conducted a graph traversal using a method called DFS, which significantly enhanced our knowledge about the behavior of graphs and their usages in both programming and everyday life situations.

**References**

[1] GeeksforGeeks, “Graph terminology in data structure,” *GeeksforGeeks*, Jul. 23, 2025. <https://www.geeksforgeeks.org/dsa/graph-terminology-in-data-structure/>

[2] “Graph Data Structure.” <https://www.programiz.com/dsa/graph>