Data Structure and Algorithm

Laboratory Activity No. 10

Intro to Graphs

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Calica, Ljay L. | Engr. Maria Rizette H. Sayo |

Oct, 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
* Undirected Graph is like a two way street, for example in Facebook friends. If you are friends with someone, they are friends with you. That’s what Undirected Graph is.
  + 1. Directed Graph
* Directed Graph is like a one way street, for example in Instagram followers. You can follow someone without them following you back.
  + 1. Nodes / 4. Vertex
* They’re same, these are just the “ dots ” in the graph. They represent things people, cities, computers, etc.
* For example, in social network. In each person is a node/vertex.

1. Degree

* It is how many connections a node has. In undirected graphs: just count the lines touching the node.
* In example, if I have 5 friends in classroom therefore the degree has 5.

1. Indegree

* Is how many arrows are pointing TO a node, and it is only in Directed Graphs.
* In example, is like how many followers I have in Facebook.

1. Outdegree

* Is how many arrows are pointing FROM a node. And it’s only in Directed Graphs.
* In example, is like how many I follow on Facebook.

1. Path

* It’s a route from one node to another, and it’s sequence of connected nodes.
* In example. You → Friend → Friend's Friend → Stranger

1. Cycle

* A path that starts and ends at the same node. It’s like a loop or circle.
* In example. You → Friend → Another Friend → You

1. Simple Cycle

* It’s a cycle where you don’t repeat any nodes (except start / end ). No detours within the loop. ( add, it’s clean loop )
* In example. You → Friend → Cousin → You

# IV.Conclusion

So, why does all this graph stuff actually matter? Whether we're scrolling through social media, using a maps app to find the quickest route, or even just browsing the internet, we're actually walking through massive, invisible graphs every single day. Those simple dots and lines we talked about? They're secretly running the digital world around us. Understanding these basics like how connections work, what direction they go in, and how to find paths is like learning the alphabet before you can write a story. It might seem technical at first, but it's the foundation that lets us build and understand the complex, connected systems that shape our modern lives.

**References**

[1] Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). Data Structures and Algorithms in Python. Wiley.

[2] Python Software Foundation. (2023). Python Documentation: Data Structures.

<https://docs.python.org/3/tutorial/datastructures.html>