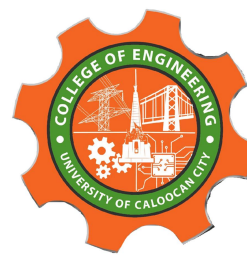




UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 10

---

# Intro to Graphs

---

*Submitted by:*  
Adoracion, Jerick Dave D.

*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.

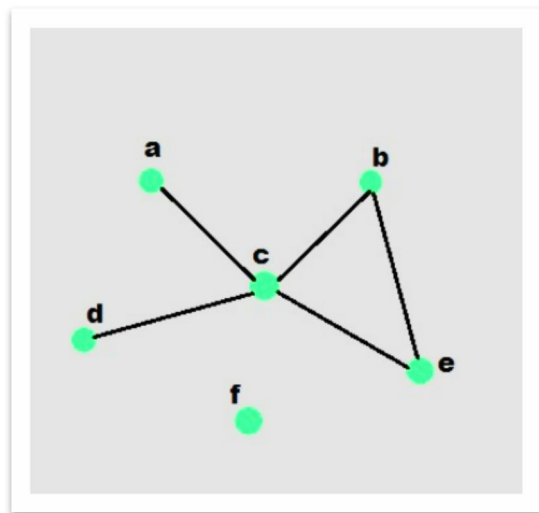


Figure 1. Sample graph with vertices and edges

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

#### **Undirected graph**

An undirected graph is a graph where the edges don't have a direction. This means the connection between two nodes works both ways. If there is an edge between node A and node B, you can travel from A to B and from B to A.

#### **Directed Graph (Digraph)**

A directed graph is one in which an arrow designates the specific direction of each edge. An edge from node A to node B does not always imply an edge from node B to node A.

#### **Nodes**

Nodes (also called vertices) are the fundamental units of a graph that represent entities or data points.

#### **Vertex**

Another name for a node in a graph is a vertex. In programming, the terms node and vertex are interchangeable.

#### **Degree**

The degree of a vertex is the number of edges connected to it. In undirected graphs, it counts all incident edges. A higher degree means the vertex is more connected.

#### **Indegree**

In a directed graph, indegree is the number of incoming edges to a vertex. It displays the number of vertices pointing to that node.

#### **Outdegree**

In a directed graph, outdegree is the number of edges that leave a vertex. It shows the number of vertices to which a node points.

#### **Path**

A series of vertices joined by edges is called a path. It displays a potential path between two vertices in a graph. Both direct and multi-intermediate vertex paths are possible.

#### **Cycle**

A cycle is a path that starts and ends at the same vertex. All other vertices in the path are distinct. It represents a loop within the graph.

#### **Simple Cycle**

A cycle that has only the beginning and ending vertices and no other repeated edges is called a simple cycle. The graph displays a single closed path.

## IV. Conclusion

In this laboratory, I've learned the fundamental ideas of graph theory, which are crucial to computer programming. Knowing the distinction between directed and undirected graphs made it easier for me to see the various ways that relationships and data can be represented. I became aware of how graphs are used to arrange and examine relationships between data points when I heard terms like nodes, vertices, and degrees.

The concepts of paths and cycles also caught my attention since they illustrate how communication or movement occurs within a network. Understanding these ideas will undoubtedly benefit me when I study data structures, algorithms, and network analysis in the future. All things considered, this lesson helped me better understand how graphs function and how practical they are for resolving real-world computing issues.

## References

- [1] Co Arthur O.. "University of Caloocan City Computer Engineering Department Honor Code," UCC-CpE Departmental Policies, 2020.
- [2] R. Diestel, *Graph Theory*, 5th ed. Berlin, Germany: Springer, 2017.
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.
- [4] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Amsterdam, Netherlands: North-Holland, 2008.
- [5] D. B. West, *Introduction to Graph Theory*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2001.