educative

What is a Graph?

This lesson is a brief introduction to the graph data structure, its types, and the standard terminologies used to describe it.

We'll cover the following

Introduction

When we talk about graphs, what comes to mind are the conventional graphs used to visualize data. In computer science, the term “graph” has a completely different meaning. It is a data structure used to store and manipulate data.

The graph data structure plays a fundamental role in several applications such as GPS, neural networks, peer-to-peer networks, search engine crawlers, garbage collection (Python), and even social networking websites.

This section will explore its functionality and power. We will also look at how they are used to solve a diverse range of problems.

Now, let’s see what a graph really is.

Graph Structure

A **graph** is a set of nodes that are connected to each other in the form of a network. First of all, we’ll define the two basic components of a graph.

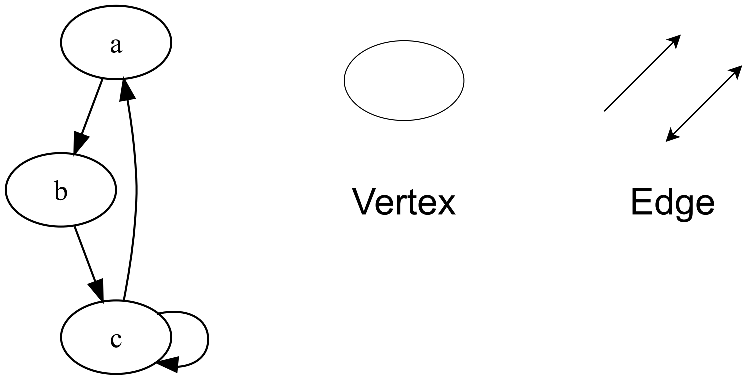
Vertex

> A **vertex** is the most essential part of a graph. A collection of vertices forms a graph. In that sense, vertices are similar to linked list nodes.

Edge

> An edge is the link between two vertices. It can be uni-directional or bi-directional depending on your graph. An edge can also have a cost associated with it (will be discussed in detail later).

Here is a visual representation of a graph:



Graph, vertices and edges

There are several terms used to describe the properties of graphs.

Graph Terminologies

1. **Degree of a Vertex**: The total number of edges incident on a vertex. There are two types of degrees:
    - **In-Degree**: The total number of incoming edges of a vertex.
    - **Out-Degree**: The total number of outgoing edges of a vertex.
2. **Parallel Edges**: Two undirected edges are parallel if they have the same end vertices. Two directed edges are parallel if they have the same starting and ending vertices.
3. **Self Loop**: This occurs when an edge starts and ends on the same vertex.
4. **Adjacency**: Two vertices are said to be adjacent if there is an edge connecting them directly.

In the illustration above, the in-degree of both **a** and **b** is 1. The same goes for the out-degree of the vertices. The in-degree and out-degree for **c** are 2 as it contains a self-loop.

In the next lesson, we'll observe the two primary types of graphs.

← Back

Next →