Graph

A graph data structure is used to represent relations between pairs of objects .

It consists of nodes (known as vertices) that are connected through links (known as edges). The relationship between the nodes can be used to model the relation between the objects in the graph. This is what makes graphs important in the real world.

Directed Graph

Directed Graph

David W.) [Publidomain]

It can be viewed as a generalization of the tree data structure as any kind of relationship can exist between the nodes of a tree, instead of the purely parent-child relationship of a tree.

Mathematically, Graph G is an ordered set (V, E) where V(G) represents the set of elements, called vertices, and E(G) represents the edges between these vertices.

A graph can be classified into 2 types:

1. Undirected Graphs

An undirected graph does not have any directed associated with its edges. This means that any edge could be traversed in both ways.

Mathematically, an edge is represented by an unordered pair [u, v] and can be traversed from u to v or vice-versa.

2. Directed Graphs

A directed graph has a direction associated with its edges. This means that any edge could be traversed only in the way of the direction.

Mathematically, an edge is represented by an ordered pair [u, v] and can only be traversed from u to v.

Basic Terminology in a graph

Vertex: An individual data element of a graph is called Vertex.

Edge: An edge is a connecting link between two vertices. An Edge is also known as Arc.

Mixed Graph: A graph with undirected and directed edges is said to be a mixed graph.

Origin: If an edge is directed, its first endpoint is said to be the origin of it.

Destination: If an edge is directed, its first endpoint is said to be the origin of it and the other endpoint is said to be the destination of the edge.

Adjacency: Two node or vertices are adjacent if they are connected through an edge.

Path: The Path represents a sequence of edges between the two vertices.

Degree: The total number of edges connected to a vertex is said to be the degree of that vertex.

In-Degree: In-degree of a vertex is the number of edges which are coming into the vertex.

Out-Degree: Out-degree of a vertex is the number of edges which are going out from the vertex.

Minimum Spanning Tree (MST): A minimum spanning tree (MST) is a subset of the edges of a connected, edge-weighted (un)directed graph that connects all the vertices, without any cycles and with the minimum possible total edge weight.

Simple Graph: A graph is said to be simple if there are no parallel and self-loop edges.

Directed acyclic graph (DAG): A directed acyclic graph (DAG) is a graph that is directed and without cycles connecting the other edges. This means that it is impossible to traverse the entire graph starting at one edge.

Weighted Graph: A weighted graph is a graph in which a number (known as the weight) is assigned to each edge. Such weights might represent for example costs, lengths or capacities, depending on the problem.

Complete Graph: A complete graph is a graph in which each pair of vertices is joined by an edge. A complete graph contains all possible edges.

Connected graph: A connected graph is an undirected graph in which every unordered pair of vertices in the graph is connected. Otherwise, it is called a disconnected graph.

Representation of a graph
A binary graph data structure can be represented using two methods:

Adjacency List Representation
In this representation, every vertex of the graph contains a linked list of its neighboring vertices and edges.

An array of lists is used where the size of the array is equal to the number of vertices. Each of the elements in the arrays contains a linked list of all the vertices adjacent to the list.

Adjacency List Representation

Adjacency List Representation

Wdvorak [CC0]

Adjacency Matrix Representation

In this representation, the graph can be represented using a matrix of size total number of vertices by the total number of vertices. Here, rows and columns both represent vertices. This matrix is filled with either 1 or 0.

Here, 1 represents there is an edge from row vertex to column vertex and 0 represents there is no edge from row vertex to a column vertex.

Adjacency Matrix Representation

Adjacency Matrix Representation

Jakkritcpenu [CC BY-SA 4.0]

To represent the weights for weighted graphs, the weight of edge (u, v) is simply stored as the entry in row u and column v of the adjacency matrix.

Adjacency matrix representation requires O(V^2) memory locations irrespective of the number of edges in the graph.

Graph Traversal Algorithms
There are two standard graph traversal algorithms:

Breadth First Search (BFS)
Depth First Search (DFS)
Breadth First Search (BFS)
Breadth-first search begins at the root node of the graph and explores all its neighbouring nodes. For each of these nodes, the algorithm again explores its neighbouring nodes. This is continued until the specified element is found or all the nodes are exhausted.

A queue is used as an auxiliary data structure to keep track of the neighboring nodes.

Breadth First Search

Breadth First Search

Braindrain0000 [Publidomain]

Depth First Search (DFS)

Depth-first search starts on a node and explores nodes going deeper and deeper until the specified node is found, or until a node with no children is found. If a node is found with no children, the algorithm backtracks and returns to the most recent node that has not been explored. This process continues until all the nodes have been traversed.

A stack is used as an auxiliary data structure to keep track of traversed nodes to help it backtrack when required.

Depth First Search

Depth First Search

Wolfram Esser [CC BY-SA 3.0]

Applications of Graphs in Programming

Family trees: can be mapped where the member nodes have an