Unit 5: Graph

Prof. A.M. Joshi

What is Graph Data Structure?

- A Graph is a non-linear data structure consisting of vertices and edges.
-
- The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph.

- More formally a Graph is composed of a set of vertices(V) and a set of edges(E).

- The graph is denoted by G(E, V).

Components of a Graph

- Vertices: Vertices are the fundamental units of the graph. Sometimes, vertices are also known as vertex or nodes. Every node/vertex can be labeled or unlabelled.

- Edges: Edges are drawn or used to connect two nodes of the graph. It can be ordered pair of nodes in a directed graph. Edges can connect any two nodes in any possible way. There are no rules. Sometimes, edges are also known as arcs. Every edge can be labeled/unlabelled.
- Applications:
- Graphs are used to solve many real-life problems. Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like linkedIn, Facebook. For example, in Facebook, each person is represented with a vertex(or node). Each node is a structure and contains information like person id, name, gender, locale etc.

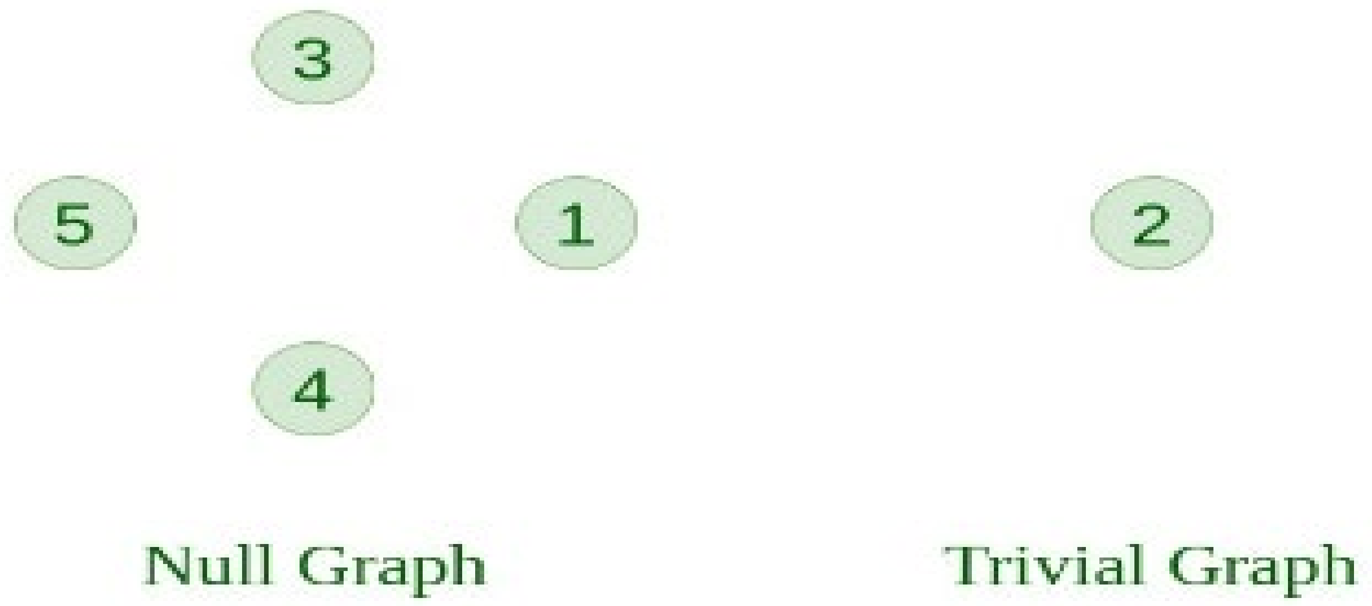
Types Of Graph

1. Null Graph

- A graph is known as a null graph if there are no edges in the graph.

2. Trivial Graph

- Graph having only a single vertex, it is also the smallest graph possible.



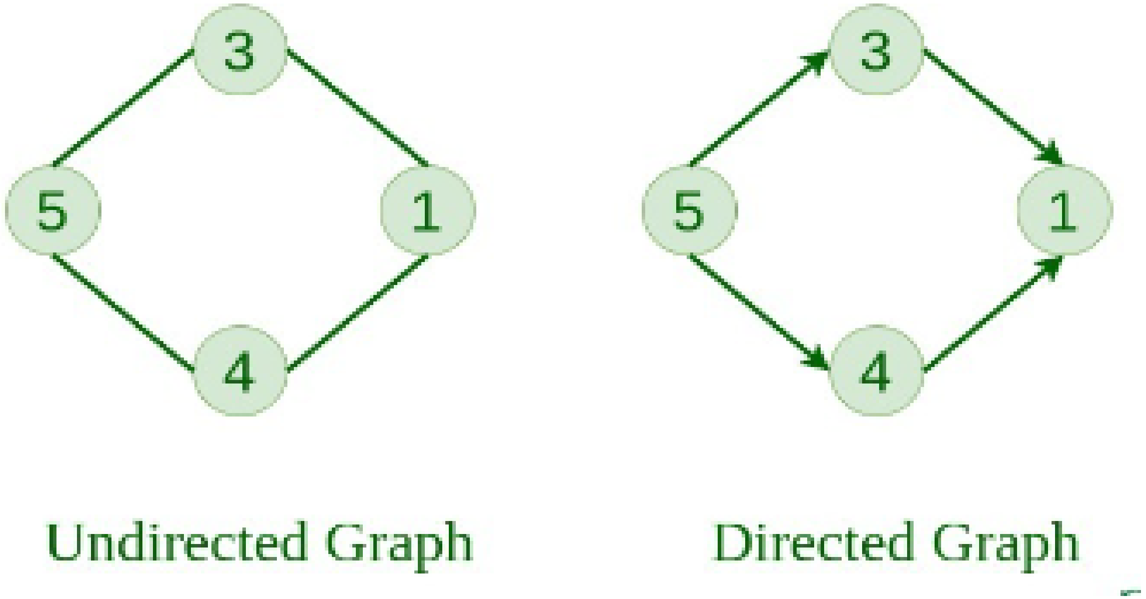
Cont...

3. Undirected Graph

- A graph in which edges do not have any direction. That is the nodes are unordered pairs in the definition of every edge.

4. Directed Graph

- A graph in which edge has direction. That is the nodes are ordered pairs in the definition of every edge.



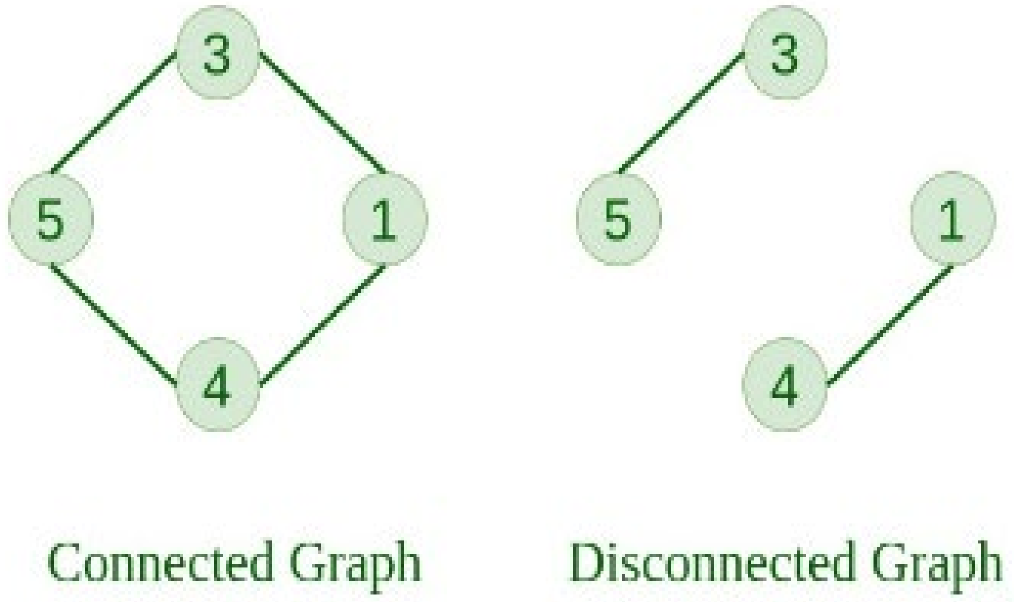
Cont...

5. Connected Graph

- The graph in which from one node we can visit any other node in the graph is known as a connected graph.

6. Disconnected Graph

- The graph in which at least one node is not reachable from a node is known as a disconnected graph.



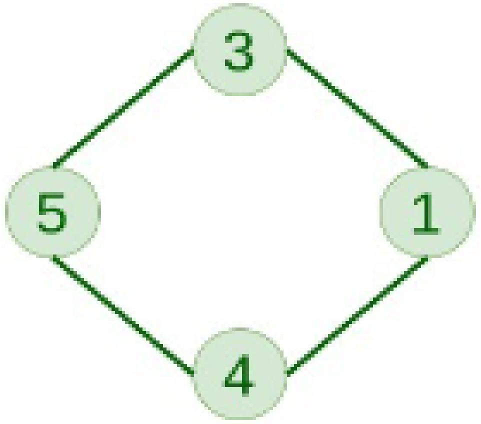
Cont...

7. Regular Graph

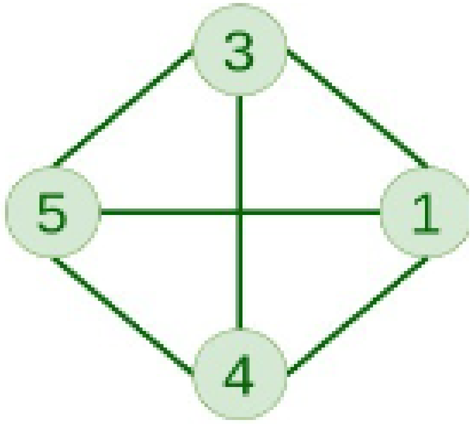
- The graph in which the degree of every vertex is equal to K is called K regular graph. It is a graph where each vertex has the same number of neighbors.

8. Complete Graph

- The graph in which from each node there is an edge to each other node.



2-Regular



Complete Graph

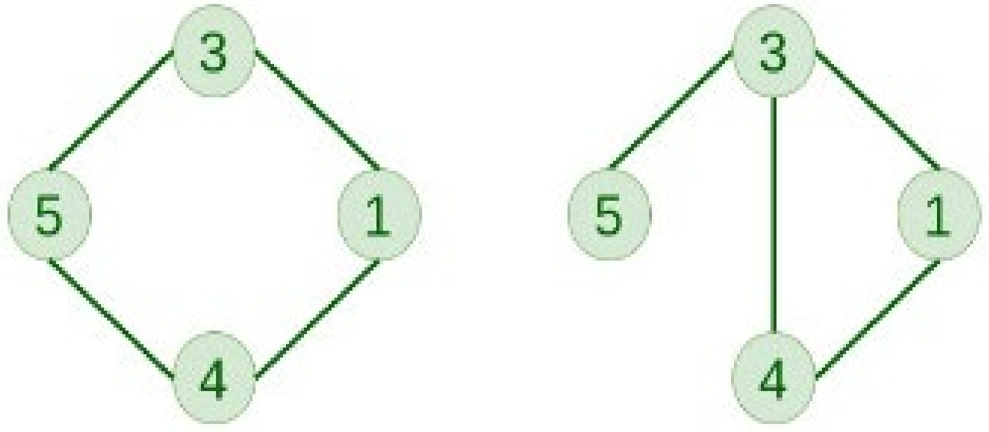
Cont...

9. Cycle Graph

- The graph in which the graph is a cycle in itself, the degree of each vertex is 2.

10. Cyclic Graph

- A graph containing at least one cycle is known as a Cyclic graph.



Cycle Graph Cyclic Graph

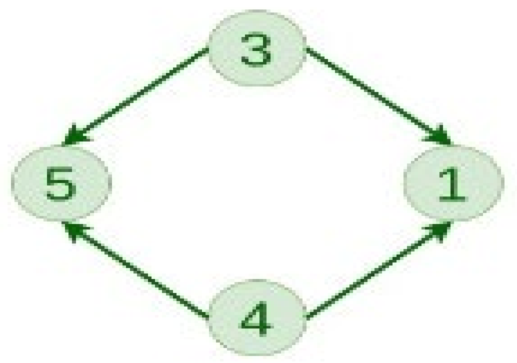
Cont...

11. Directed Acyclic Graph

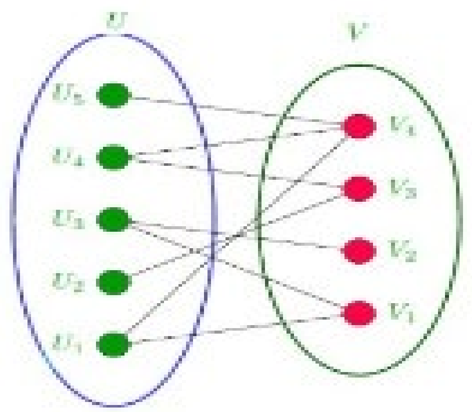
- A Directed Graph that does not contain any cycle.

12. Bipartite Graph

- A graph in which vertex can be divided into two sets such that vertex in each set does not contain any edge between them.



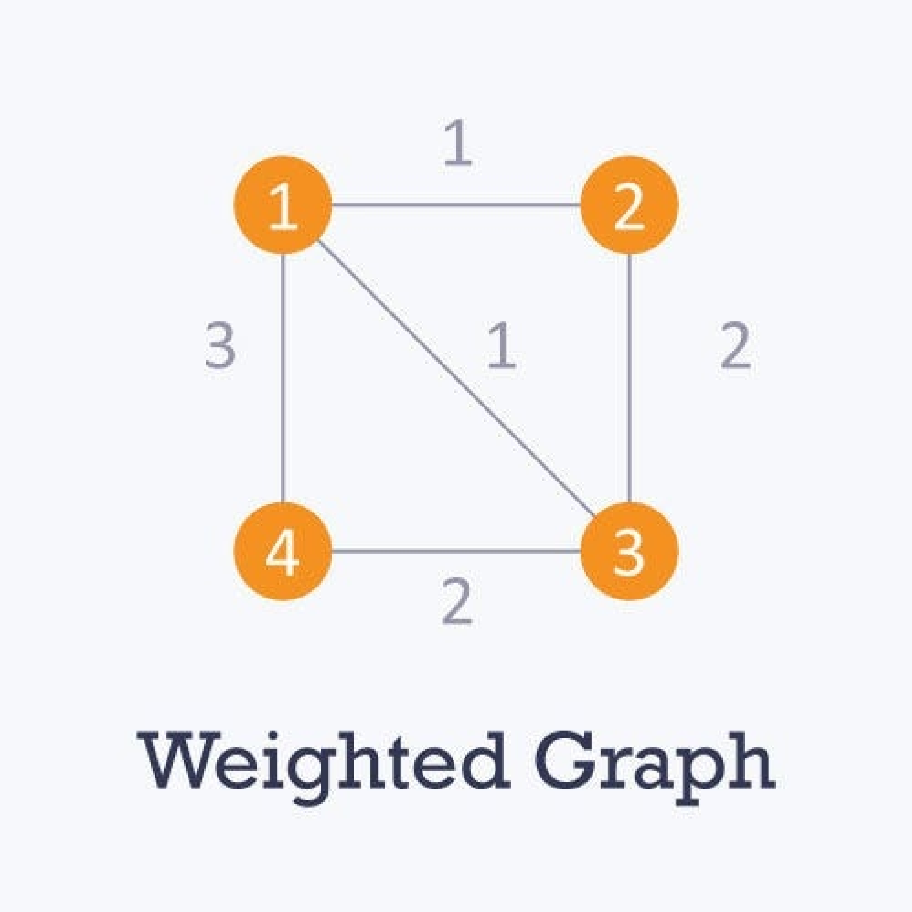
Directed Acyclic Graph


Bipartite Graph

Cont...

13. Weighted Graph

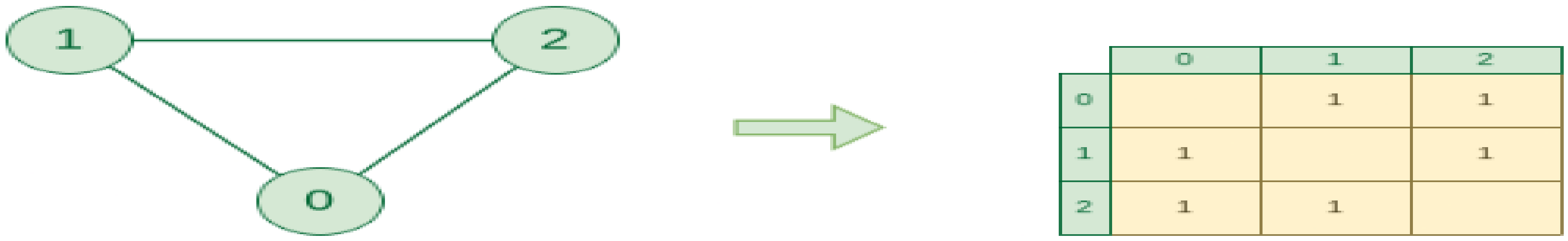
- A graph in which the edges are already specified with suitable weight is known as a weighted graph.
- Weighted graphs can be further classified as directed weighted graphs and undirected weighted graphs.



Representation of Graphs

- There are two ways to store a graph:
- Adjacency Matrix
- Adjacency List

- Adjacency Matrix for Undirected Graph.
- The below figure shows an undirected graph. Initially, the entire Matrix is initialized to 0. If there is an edge from source to destination, we insert 1 to both cases (adjMat[destination] and adjMat[destination]) because we can go either way.

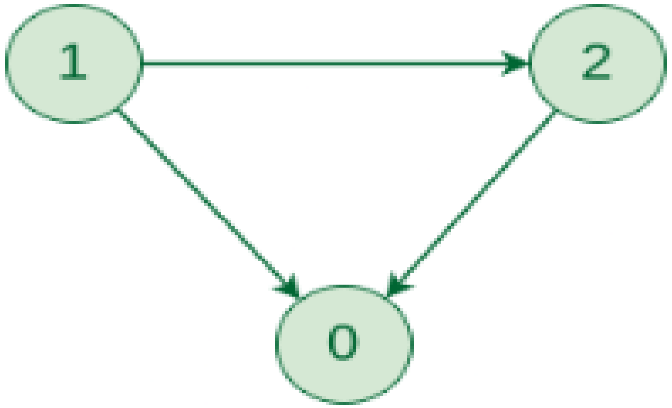


Undirected Graph Adjacency Matrix

Graph Representation of Undirected graph to Adjacency Matrix

Representation of Directed Graph to Adjacency Matrix:

- The below figure shows a directed graph.
- Initially, the entire Matrix is initialized to 0.
- If there is an edge from source to destination, we insert 1 for that particular adjMat[destination].



Directed Graph

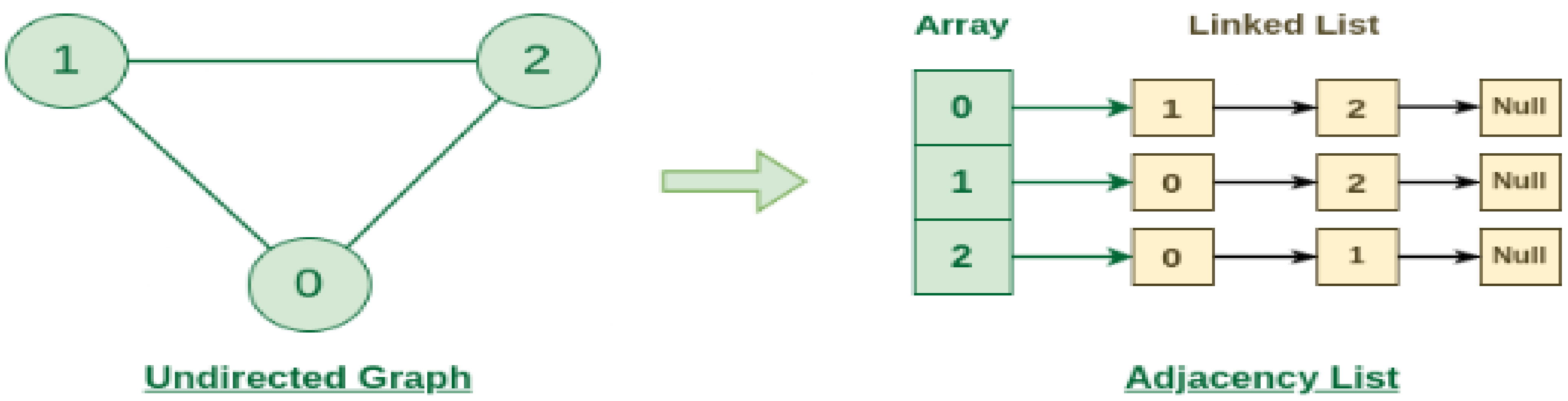
	0	1	2
0			
1	1		1
2	1		

Adjacency Matrix

Graph Representation of Directed graph to Adjacency Matrix

Cont...

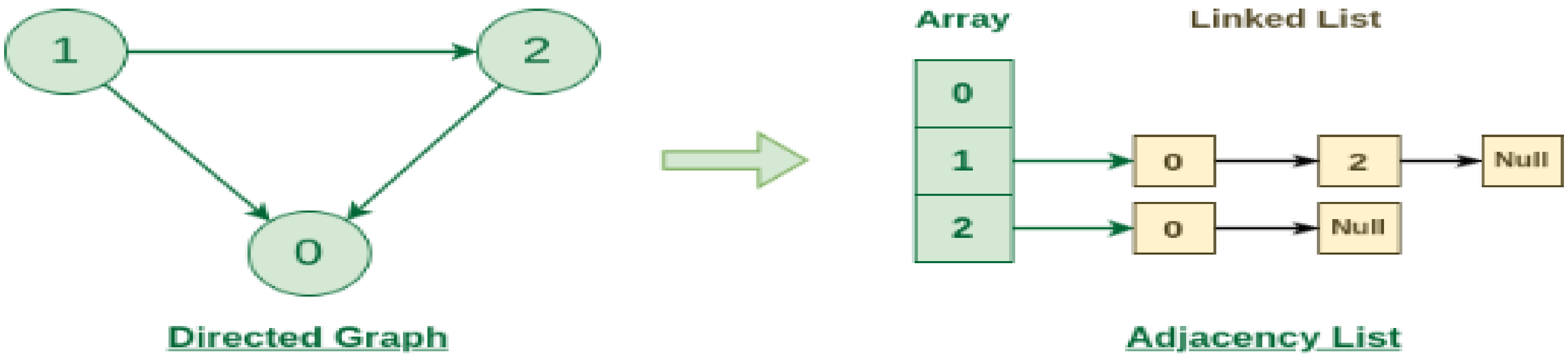
- Adjacency List
- This graph is represented as a collection of linked lists.
- There is an array of pointer which points to the edges connected to that vertex.
- Representation of Undirected Graph to Adjacency list:



Graph Representation of Undirected graph to Adjacency List

Representation of Directed Graph to Adjacency list:

- The below directed graph has 3 vertices. So, an array of list will be created of size 3, where each indices represent the vertices.
- Now, vertex 0 has no neighbours. For vertex 1, it has two neighbour (i.e, 0 and 2) So, insert vertices 0 and 2 at indices 1 of array.
- Similarly, for vertex 2, insert its neighbours in array of list.



Graph Representation of Directed graph to Adjacency List

Graph Traversal

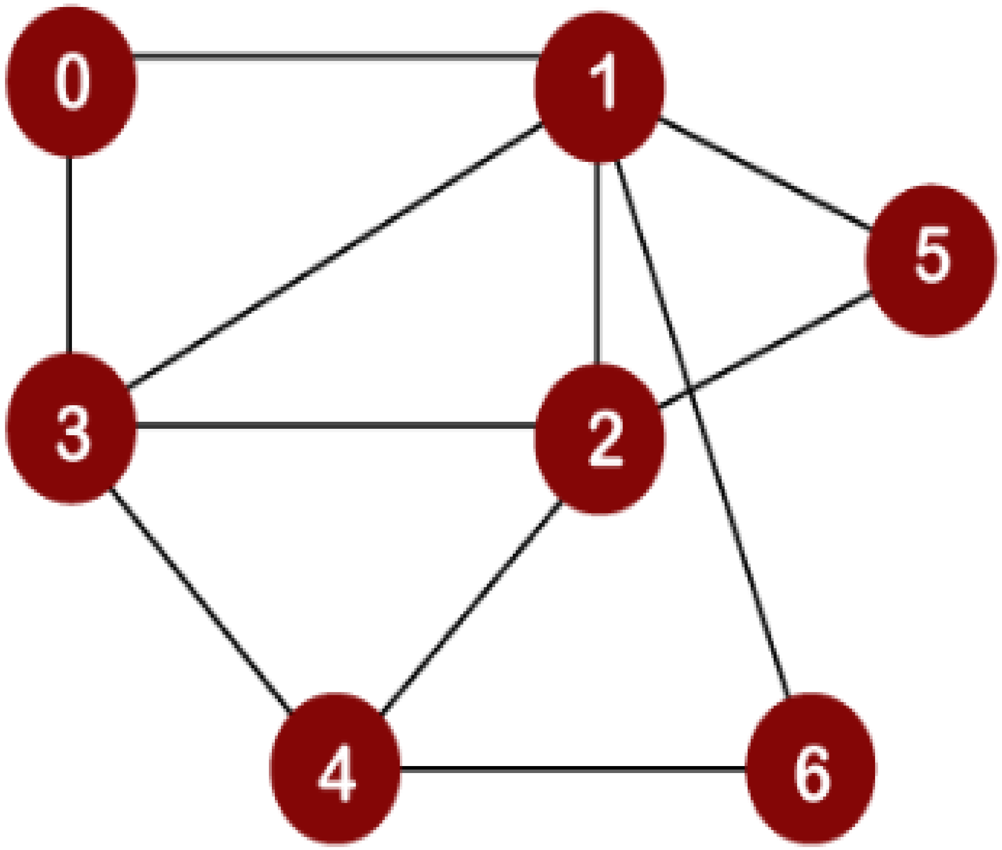
- Graph traversal is visiting or updating each vertex in a graph. The order in which they visit the vertices classifies the traversals. There are two ways to implement a graph traversal:

- Breadth-First Search (BFS) – It is a traversal operation that horizontally traverses the graph. It traverses all the nodes at a single level before moving to the next level. It begins at the graph's root and traverses all the nodes at a single depth level before moving on to the next level.

- Depth-First Search (DFS): This is another traversal operation that traverses the graph vertically. It starts with the root node of the graph and investigates each branch as far as feasible before backtracking.

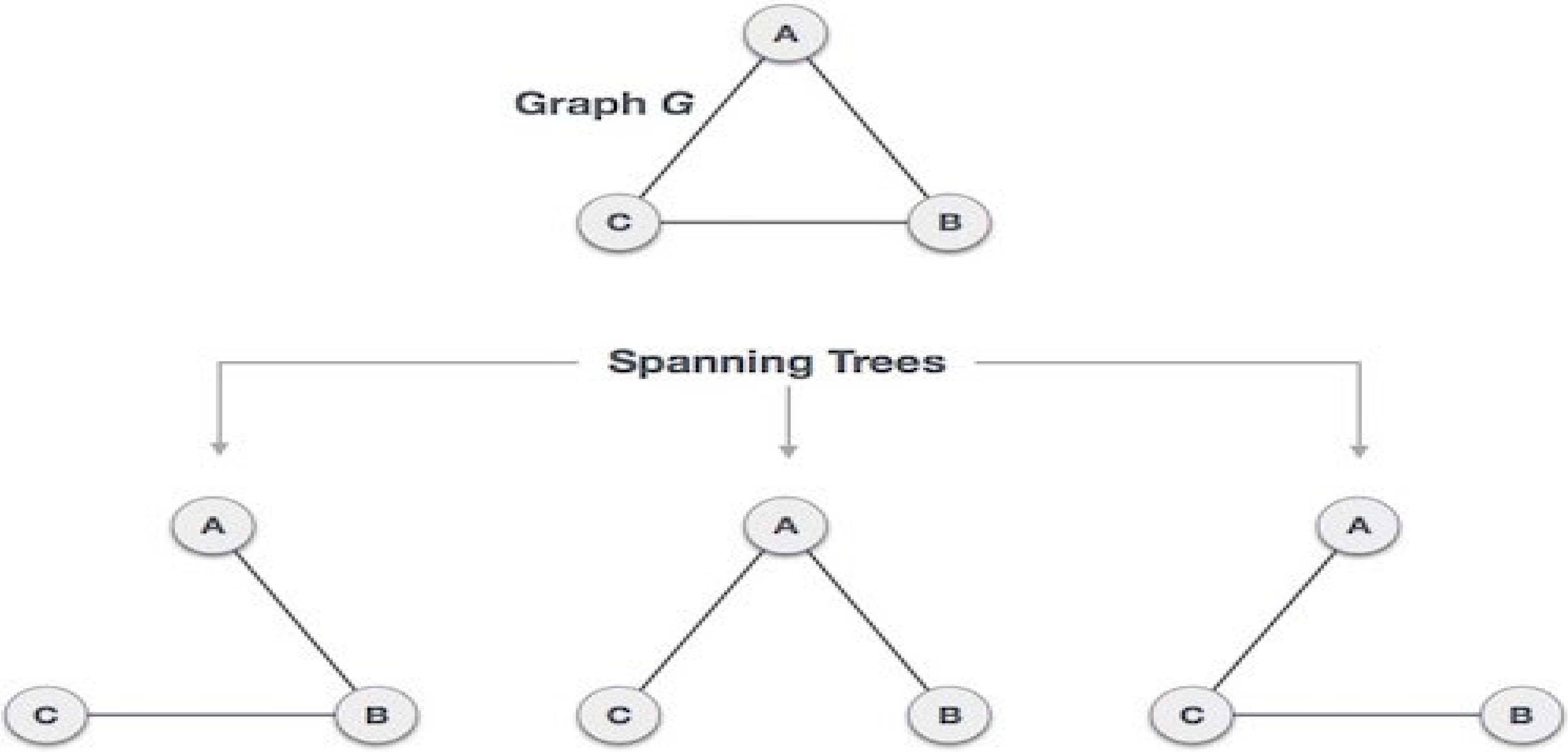
What is BFS?

- BFS stands for Breadth First Search.
- It is also known as level order traversal.
- The Queue data structure is used for the Breadth First Search traversal.
- When we use the BFS algorithm for the traversal in a graph, we can consider any node as a root node.
- Let's consider the below graph for the breadth first search traversal.



Spanning Tree

- A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges.
- Hence, a spanning tree does not have cycles and it cannot be disconnected.
- By this definition, we can draw a conclusion that every connected and undirected Graph G has at least one spanning tree.
- A disconnected graph does not have any spanning tree, as it cannot be spanned to all its vertices.
- We found three spanning trees off one complete graph.
- A complete undirected graph can have maximum nn-2 number of spanning trees, where n is the number of nodes.
- In the above addressed example, n is 3, hence 3^3-2 spanning trees are possible.



Example: Spanning Tree

General Properties of Spanning Tree

- Following are a few properties of the spanning tree connected to graph G −
- A connected graph G can have more than one spanning tree.
- All possible spanning trees of graph G, have the same number of edges and vertices.
- The spanning tree does not have any cycle (loops).
- Removing one edge from the spanning tree will make the graph disconnected, i.e. the spanning tree is minimally connected.
- Adding one edge to the spanning tree will create a circuit or loop, i.e. the spanning tree is maximally acyclic.

Minimum Spanning Tree (MST)

- In a weighted graph, a minimum spanning tree is a spanning tree that has minimum weight than all other spanning trees of the same graph.
- In real-world situations, this weight can be measured as distance, congestion, traffic load or any arbitrary value denoted to the edges.

- Minimum Spanning-Tree Algorithm
- We shall learn about two most important spanning tree algorithms here −

- Kruskal's Algorithm

- Prim's Algorithm