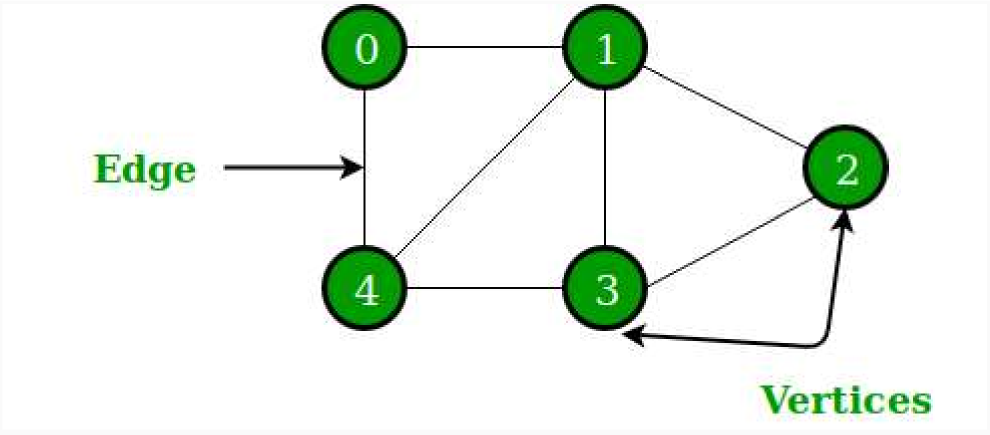
Graph Data Structure And Algorithms

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

A Graph consists of a finite set of vertices(or nodes) and a set of Edges which connect a pair of nodes.



In the above Graph, the set of vertices V = {0,1,2,3,4} and the set of edges E = {01, 12, 23, 34, 04, 14, 13}.

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.

The graph has two types of traversal algorithms. These are called the Breadth First Search and Depth First Search.

Breadth First Search (BFS)

The Breadth First Search (BFS) traversal is an algorithm, which is used to visit all of the nodes of a given graph. In this traversal algorithm one node is selected and then all of the adjacent nodes are visited one by one. After completing all of the adjacent vertices, it moves further to check another vertices and checks its adjacent vertices again.

Algorithm

```
bfs(vertices, start)

Input: The list of vertices, and the start vertex.

Output: Traverse all of the nodes, if the graph is connected.

Begin

   define an empty queue que

   at first mark all nodes status as unvisited

   add the start vertex into the que

   while que is not empty, do

      delete item from que and set to u

      display the vertex u

      for all vertices 1 adjacent with u, do
```

```
         if vertices[i] is unvisited, then
            mark vertices[i] as temporarily visited
            add v into the queue
         mark
      done
      mark u as completely visited
   done
End
```

Depth First Search (DFS)

The Depth First Search (DFS) is a graph traversal algorithm. In this algorithm one starting vertex is given, and when an adjacent vertex is found, it moves to that adjacent vertex first and try to traverse in the same manner.

Algorithm

```
dfs(vertices, start)
Input: The list of all vertices, and the start node.
Output: Traverse all nodes in the graph.
Begin
   initially make the state to unvisited for all nodes
   push start into the stack
   while stack is not empty, do
```

```
      pop element from stack and set to u
      display the node u
      if u is not visited, then
         mark u as visited
         for all nodes i connected to u, do
            if ith vertex is unvisited, then
               push ith vertex into the stack
               mark ith vertex as visited
         done
   done
End
```