Graphs:

Graphs are composed of edges and nodes. A node is data object that may have specific values for each of its features. A edge or link shows a relationship or a connection between two nodes. One other way to represent an edge is the ability it provides to visit a neighboring node based on its bi-directional or uni-directional properties.

Types of graphs:

*Communication networks*

A network of computers connecting together in different graph orientations or topologies. A node is a representation of a computer or processing point, and an edge represents a physical link connecting the two processors.

*Transportation networks*

The airline flight network consists of airports (nodes). A valid edge represents a flight from one airport connecting to another airport.

*Information networks:*

Consider the www as an information network the nodes (u,v) are web pages. The page u connects to v if u has a hyperlink to v.

*Social Networks*

This graph consists of people, in which each link from one person to another represents a friend relationship. The degree of separation can determine if one person can know another person using the transitive property of friendship. This property is shown more apparently through a graph.

*Dependency networks*

Within an ecosystem there are dependency relations (u,v) where species u eats species v but v is not the predator to u. Compilation scripts use dependency graphs where a certain compilation will not execute until another compilation executes first.

Edge properties.

A graph can be directed or undirected. For connecting nodes. For a undirected graph, neighboring nodes can visit/relate with each other. For example John is a friend of Mark and Mark is a friend of John. Directed graphs are composed of directed edges. With a directed edge e.g. (A -> B) A can reach B but B cannot reach An edge can also have weights. A weight can represent the difficulty factor of reaching a connecting neighbor node. An example of this representation is through a graph of airplane routes from one city to another. Certain flight connections may have a higher weight due to the complications e.g distance, weather conditions between the connecting cities.

*Graph properties*

Every graph has a root or head node, this is the starting point of a graph. Just like a family tree, the links under a node are the children. Each link down, is denoted as a level. Any nodes that do not have any links under it are the leaf nodes. The depth of a graph is based on the path with the maximum number of links from the head node to a leaf node.

Stack and Queues

Having a stack and queue data structure is integral for traversing through a graph.

A stack puts each incoming data object at the beginning of the list (LIFO) last in first out. A queue puts each incoming data object at the end of the list (FIFO) first in, first out.

Breadth first search

A breadth first search visits each level of the tree starting from the top, to the leaf level.

A breadth first search traverses the graph by popping off the head node and queuing its unvisited neighbors. This process is continued until the queue is empty (all the nodes have been visited).

Depth first search

A depth first search visits each node down the path of a tree to the leaf node. When a traversal cannot go down any further the algorithm goes up one level of the tree and continues down another path that has not been visited. This same algorithm works by popping the head node and adding the unvisited neighbors to the top of the stack.