Graph Traversal

Breadth First Search

Name : BFSearch
Given : Graph G represented by adjacency list
Start, Starting Vertex
Returns: Nothing
Variables: Array Visited of size equal to number of vertices
v: number to specify a vertex
Current : to store address of a node in the adjacency list
Q: queue to store vertices

Steps:

1) Initialise q to empty status.
2) Mark every vertex as not visited by initializing the elements of array Visited.
3) Mark vertex start as visited
4) Add vertex start to the queue
5) While the queue q is not empty
    a) Delete the front node from q and store the vertex deleted in v
    b) Print vertex v
    c) Current = first node in the adjacency list for vertex v
    d) While current is not NULL
        i) V = vertex contained in current
        ii) If vertex v is not yet visited
            (a) Add vertex v to the queue q
            (b) Mark vertex v as visited
        iii) Current = next of current

Depth First Search

Name : DFSearch
Given : Graph G represented by adjacency list
Start, Starting Vertex
Returns: Nothing
Variables: Array Visited of size equal to number of vertices
v: number to specify a vertex
Current : pointer to a node in the adjacency list
S: stack to store vertices

Steps:

1) Initialise S to empty status.
2) Mark every vertex as not visited by initializing the elements of array Visited.
3) Mark vertex start as visited
4) Push vertex start on to the stack
5) While the stack S is not empty
    a) Pop the stack S and store the vertex deleted in v
    b) Print vertex v
    c) Current = first node in the adjacency list for vertex v
    d) While current is not NULL
        i) V = vertex contained in current
        ii) If vertex v is not yet visited
            (a) Push vertex v on to the stack S
            (b) Mark vertex v as visited
        iii) Current = next of current