**Week 16: Graph**

**Aim:** Implement the following Traversals on the given Graph.

[a] BFS

**Program:**

#include <iostream>

#include <queue>

#include <vector>

using namespace std;

const int MAXN = 100; // maximum number of nodes

vector<int> adj[MAXN]; // adjacency list of graph

bool vis[MAXN]; // visited array to keep track of which nodes have been visited

void bfs(int startNode) {

queue<int> q;

q.push(startNode);

vis[startNode] = true;

while (!q.empty()) {

int currNode = q.front();

q.pop();

cout << currNode << " ";

for (int i = 0; i < adj[currNode].size(); i++) {

int nextNode = adj[currNode][i];

if (!vis[nextNode]) {

q.push(nextNode);

vis[nextNode] = true;

}

}

}

}

int main() {

// initialize graph

adj[0].push\_back(1);

adj[0].push\_back(2);

adj[1].push\_back(2);

adj[2].push\_back(0);

adj[2].push\_back(3);

adj[3].push\_back(3);

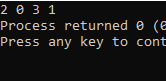
// run bfs from node 2

bfs(2);

return 0;

}

**Input & Output:**

****

[b] DFS

**Program:**

#include <iostream>

#include <stack>

#include <vector>

using namespace std;

const int MAXN = 100; // maximum number of nodes

vector<int> adj[MAXN]; // adjacency list representation

bool visited[MAXN]; // keep track of visited nodes

void dfs(int start) {

stack<int> st;

st.push(start);

while (!st.empty()) {

int node = st.top();

st.pop();

if (visited[node]) continue; // already visited

visited[node] = true;

cout << "Visited node " << node << endl;

for (int i = 0; i < adj[node].size(); i++) {

int neighbor = adj[node][i];

if (!visited[neighbor]) {

st.push(neighbor);

cout << " Visit edge (" << node << ", " << neighbor << ")" << endl;

}

}

}

}

int main() {

// create a graph

adj[0].push\_back(1);

adj[0].push\_back(2);

adj[1].push\_back(2);

adj[2].push\_back(0);

adj[2].push\_back(3);

adj[3].push\_back(3);

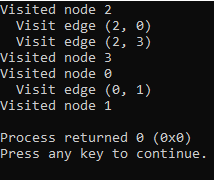
// start DFS from node 2

dfs(2);

return 0;

}

**Input & Output:**

****

**Conclusion:**

We studied about Graph. A graph is a non-linear kind of data structure made up of nodes or vertices and edges. The edges connect any two nodes in the graph, and the nodes are also known as vertices. BFS is a traversal technique in which all the nodes of the same level are explored first, and then we move to the next level. DFS is also a traversal technique in which traversal is started from the root node and explore the nodes as far as possible until we reach the node that has no unvisited adjacent nodes.