

A picture containing logo

Description automatically generated

Shahid Hasan Shuvo

ID: 221002038

CSE 206

Dr. Faiz Al Faisal

Implementation of DFS Algorithm

Green University of Bangladesh

Paper Due Date  
November 07, 2023

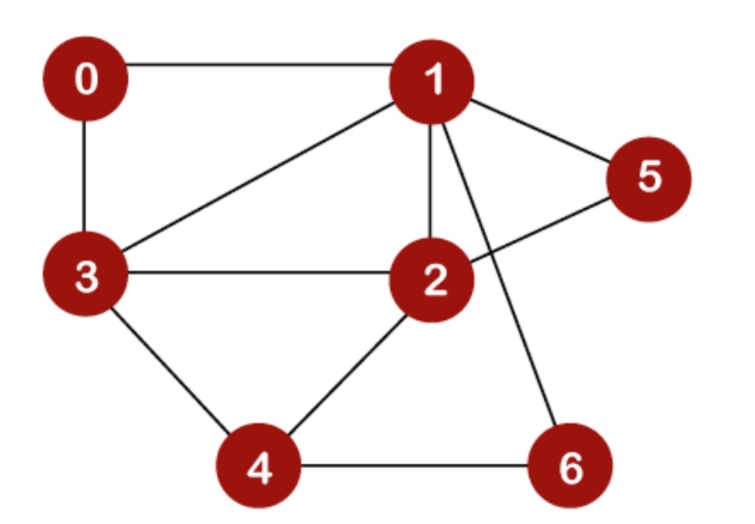
**CLP - 01**

**Experiment Name:**

Implement DFS algorithm.

a. Create an Adjacent Matrix to take input the following graph.

b. Calculate the start time and finish time for the complete graph.



**Source Code:**

#include <stdio.h>

#include <stdlib.h>

#define maxVertices 7

int adjacencyMatrix[maxVertices][maxVertices];

int visited[maxVertices];

int startTime[maxVertices];

int finishTime[maxVertices];

int currentTime = 0;

void enqueue(int vertex) {

    visited[vertex] = 1;

    startTime[vertex] = ++currentTime;

    for (int i = 0; i < maxVertices; i++) {

        if (adjacencyMatrix[vertex][i] == 1 && !visited[i]) {

            enqueue(i);

        }

    }

    finishTime[vertex] = ++currentTime;

}

void inputGraph(int vertices) {

    printf("Enter the adjacency matrix for the graph:\n");

    for (int i = 0; i < vertices; i++) {

        for (int j = 0; j < vertices; j++) {

            scanf("%d", &adjacencyMatrix[i][j]);

        }

    }

}

int main() {

    int vertices = maxVertices;

    for (int i = 0; i < vertices; i++) {

        visited[i] = 0;

    }

    inputGraph(vertices);

    printf("Depth-First Search Start and Finish Times:\n");

    for (int i = 0; i < vertices; i++) {

        if (!visited[i]) {

            enqueue(i);

        }

    }

    for (int i = 0; i < vertices; i++) {

        printf("Vertex %d: Start Time = %d, Finish Time = %d\n", i, startTime[i], finishTime[i]);

    }

    return 0;

}

    /\*

    0 1 0 1 0 0 0

    1 0 1 1 0 1 1

    0 1 0 1 1 1 0

    1 1 1 0 1 0 0

    0 0 1 1 0 0 1

    0 1 1 0 0 0 0

    0 1 0 0 1 0 0

    \*/

**Output:**

