

Institute of Computer Technology
B. Tech. Computer Science and Engineering
Sub: DS
Course Code: 2CSE302

Practical – 18

Name: Jaymin Gondaliya

Enrollment No: 23162171007

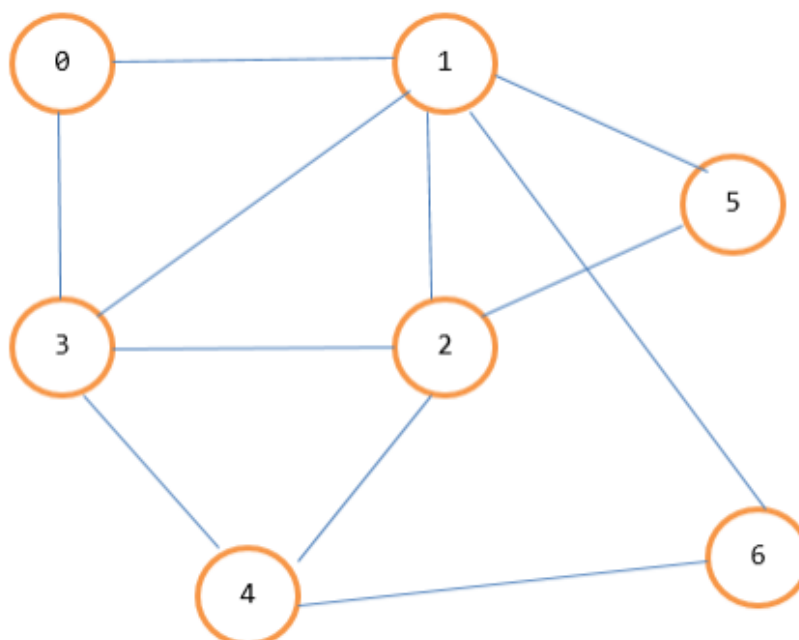
Sem - 3

Branch: CS

Class: A

Batch: 32

Using breadth first Search and Depth First Search algorithms print the traversal of the graph from all the vertex.



Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
```

```
#define MAX 7 // Number of vertices in the graph

// Adjacency matrix representation of the graph
int graph[MAX][MAX] = {
    {0, 1, 0, 1, 0, 0, 0}, // Connections from vertex 0
    {1, 0, 1, 1, 0, 1, 0}, // Connections from vertex 1
    {0, 1, 0, 1, 0, 0, 1}, // Connections from vertex 2
    {1, 1, 1, 0, 1, 0, 0}, // Connections from vertex 3
    {0, 0, 0, 1, 0, 0, 1}, // Connections from vertex 4
    {0, 1, 0, 0, 0, 0, 1}, // Connections from vertex 5
    {0, 0, 1, 0, 1, 1, 0}  // Connections from vertex 6
};

// BFS Function
void bfs(int startVertex) {
    bool visited[MAX] = {false}; // Track visited vertices
    int queue[MAX];              // Queue for BFS
    int front = 0, rear = 0;     // Queue indices

    // Start with the given vertex
    visited[startVertex] = true;
    queue[rear++] = startVertex;

    printf("BFS Traversal: ");
    while (front < rear) {
        int currentVertex = queue[front++];
        printf("%d ", currentVertex);

        // Enqueue all adjacent vertices of the current vertex
        for (int i = 0; i < MAX; i++) {
            if (graph[currentVertex][i] && !visited[i]) {
                visited[i] = true;
                queue[rear++] = i;
            }
        }
    }
    printf("\n");
}

// DFS Helper Function
void dfsUtil(int vertex, bool visited[]) {
    visited[vertex] = true;
    printf("%d ", vertex);

    // Visit all adjacent vertices of the current vertex
    for (int i = 0; i < MAX; i++) {
        if (graph[vertex][i] && !visited[i]) {
```

```
        dfsUtil(i, visited);
    }
}

// DFS Function
void dfs(int startVertex) {
    bool visited[MAX] = {false}; // Track visited vertices

    printf("DFS Traversal: ");
    dfsUtil(startVertex, visited);
    printf("\n");
}

int main() {
    int startVertex = 0; // Starting vertex for traversals

    // Perform BFS and DFS
    bfs(startVertex);
    dfs(startVertex);

    return 0;
}
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR  COMMENTS

● PS C:\ICT\SEM-3\DS\Practical> cd 'c:\ICT\SEM-3\DS\Practical\Practical-18\output'
● PS C:\ICT\SEM-3\DS\Practical\Practical-18\output> & .\'main.exe'
  BFS Traversal: 0 1 3 2 5 4 6
  DFS Traversal: 0 1 2 3 4 6 5
○ PS C:\ICT\SEM-3\DS\Practical\Practical-18\output> |
```