Design and Analysis of Algorithms

Week-3 Assignment

1. BFS

PROGRAM:

```
#include <stdio.h>

#define MAX 10

int queue[MAX], front = -1, rear = -1;
int visited[MAX];

// Function to insert into queue
void enqueue(int v) {
    if (rear == MAX - 1)
        return;
    if (front == -1)
        front = 0;
    queue[++rear] = v;
}

// Function to delete from queue
int dequeue() {
    if (front == -1 || front > rear)
        return -1;
    return queue[front++];
}

// BFS function
void bfs(int n, int adj[MAX][MAX], int start) {
    int i, v;
    enqueue(start);
    visited[start] = 1;

    while (front <= rear) {
        v = dequeue();
        printf("%d ", v);

        for (i = 0; i < n; i++) {
```

```
            if (adj[v][i] == 1 && !visited[i]) {
                enqueue(i);
                visited[i] = 1;
            }
        }
    }
}

int main() {
    int n, i, j, start;
    int adj[MAX][MAX];

    printf("Enter number of vertices: ");
    scanf("%d", &n);

    printf("Enter adjacency matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &adj[i][j]);
        }
        visited[i] = 0;
    }

    printf("Enter starting vertex: ");
    scanf("%d", &start);

    bfs(n, adj, start);

    return 0;
}
```

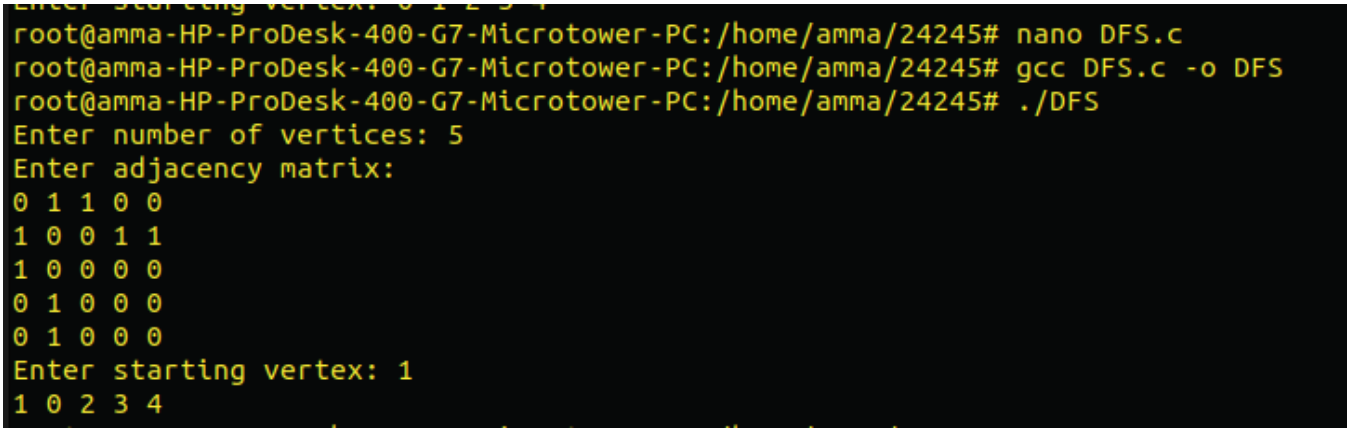
```
root@amma-HP-ProDesk-400-G7-Microtower-PC:/home/amma/24245#
root@amma-HP-ProDesk-400-G7-Microtower-PC:/home/amma/24245# nano BFS.c
root@amma-HP-ProDesk-400-G7-Microtower-PC:/home/amma/24245# gcc BFS.c -o BFS
root@amma-HP-ProDesk-400-G7-Microtower-PC:/home/amma/24245# ./BFS
Enter number of vertices: 4
Enter adjacency matrix:
0 1 1 0
1 0 0 1
1 0 0 0
1 0 0 0
Enter starting vertex: 0
0 1 2 3
```

2.DFS

PROGRAM:

```
#include <stdio.h>


#define MAX 10


int adj[MAX][MAX];
int visited[MAX];
int n;


// DFS function
void dfs(int v) {
    int i;
    printf("%d ", v);
    visited[v] = 1;


    for (i = 0; i < n; i++) {
        if (adj[v][i] == 1 && !visited[i]) {
            dfs(i);
        }
```

```
    }
}

int main() {
    int i, j, start;

    printf("Enter number of vertices: ");
    scanf("%d", &n);

    printf("Enter adjacency matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &adj[i][j]);
        }
        visited[i] = 0;
    }

    printf("Enter starting vertex: ");
    scanf("%d", &start);

    dfs(start);

    return 0;
}
```



DONIRAJ SALINDRA
CH.SC.U4CSE24245