DAA Lab3

Yugendra Senthil Kumar

CSE-C

CH.SC.U4CSE24252

1.DFS

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

int *adj;
int *visited;
int v;

void dfs(int start){
    printf("%d ", start);
    visited[start] = 1;

    for(int i = 0; i < v; ++i){
        if(adj[start * v + i] == 1 && visited[i] == 0){
            dfs(i);
        }
    }
}

int main(){
    int start;

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

    adj = (int *)malloc(v * v * sizeof(int));
    visited = (int *)malloc(v * sizeof(int));

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

```
    printf("Enter starting vertex\n");
    scanf("%d", &start);

    for(int i = 0; i < v; ++i){
        visited[i] = 0;
    }

    printf("DFS traversal\n");
    dfs(start);

    free(adj);
    free(visited);
}
```

```
yugen@fountain:/mnt/c/Users/yugen/C files/Lab3$ ./dfs
Enter number of vertices
4
Enter adjacency matrix
0 1 1 0
1 0 1 0
1 1 0 1
0 0 1 0
Enter starting vertex
0
DFS traversal
0 1 2 3 yugen@fountain:/mnt/c/Users/yugen/C files/Lab3$
```

2.BFS

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

int *adj;
int *visited;
int v;

void bfs(int start){
    int *queue;
    int front = 0, rear = 0;

    queue = (int *)malloc(v * sizeof(int));
```

```
    visited[start] = 1;
    queue[rear++] = start;

    while(front < rear){
        int current = queue[front++];
        printf("%d ", current);

        for(int i = 0; i < v; ++i){
            if(adj[current * v + i] == 1 && visited[i] == 0){
                visited[i] = 1;
                queue[rear++] = i;
            }
        }
    }

    free(queue);
}

int main(){
    int start;

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

    adj = (int *)malloc(v * v * sizeof(int));
    visited = (int *)malloc(v * sizeof(int));

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

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

    for(int i = 0; i < v; ++i){
        visited[i] = 0;
    }

    printf("BFS traversal\n");
```

```
    bfs(start);

    free(adj);
    free(visited);
}
```

```
0 1 2 3 yugen@fountain:/mnt/c/Users/yugen/C files/Lab3$ ./bfs
Enter number of vertices
5
Enter adjacency matrix
0 1 1 0 0
1 0 0 1 0
1 0 0 0 1
0 1 0 0 0
0 0 1 0 0
Enter starting vertex
0
BFS traversal
0 1 2 3 4 yugen@fountain:/mnt/c/Users/yugen/C files/Lab3$
```