WEEK - 3

(Design and Analysis Algorithms)

NAME: KANTAMRAJU SREEJA

ROLL NO: CH.SC.U4CSE24255

1Q) BFS (BREADTH FIRST SEARCH)

CODE:

```c
#include <stdio.h>
int queue[20], front = -1, rear = -1;
int visited[20];
int graph[20][20];
int n;
void bfs(int start)
{
int i;
queue[++rear] = start;
visited[start] = 1;
while (front != rear)
{
start = queue[++front];
printf("%d ", start);
for (i = 0; i < n; i++)
{
if (graph[start][i] == 1 && visited[i] == 0)
{
queue[++rear] = i;
visited[i] = 1;
}}}}
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", &graph[i][j]);
printf("Enter starting vertex: ");
scanf("%d", &start);
printf("BFS traversal: ");
bfs(start);
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc bfs.c -o bfs
kantamraju@kantamraju-VirtualBox:~$ ./bfs
Enter number of vertices: 3
Enter adjacency matrix:
1 0 1
0 1 0
0 0 1
Enter starting vertex: 0
BFS traversal: 0 2
```

2Q) DFS (DEPTH FIRST SEARCH)

CODE:

```c
#include <stdio.h>
int graph[20][20];
int visited[20];
int n;
void dfs(int start)
{
int i;
printf("%d ", start);
visited[start] = 1;
for (i = 0; i < n; i++)
{
if (graph[start][i] == 1 && visited[i] == 0)
{
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", &graph[i][j]);
printf("Enter starting vertex: ");
scanf("%d", &start);
printf("DFS traversal: ");
dfs(start);
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc dfs.c -o dfs
kantamraju@kantamraju-VirtualBox:~$ ./dfs
Enter number of vertices: 3
Enter adjacency matrix:
1 0 1
0 1 0
0 0 1
Enter starting vertex: 0
DFS traversal: 0 2
```