program 12: BST & DST

Date
Page

a) ~~Binary Search~~ Breadth First Search

b) Depth First Seach

pseudocode

```
void BFS (int start) {
        while (front < rear) {
        int node = queue[front++];
        for (int i=0; i<n; i++) {
          if (graph[node][i] == 1 && !visited[i]) {
            visited[i] = 1;
            queue[rear++] = i; } } }

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

code:

```
#include <stdio.h>
int graph[20][20], visited[20], n;

void BFS (int start) {
    int queue[20], front = 0, rear = 0;
    visited[start] = 1;
    queue[rear++] = start;

    while (front < rear) {
        int node = queue[front++];
        printf ("%d ", node);
        for (int i = 0; i < n; i++) {
            if (graph[node][i] == 1 && ! visited[i]) {
```

```
            visited[i] = 1;
            queue[rear++] = i;
        }
    }
}

int main(){
    int start;
    printf("enter number of vertices: ");
    scanf("%d", &n);
    printf("enter adjacency matrix: \n");
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            scanf("%d", &graph[i][j]);
    for(int i=0; i<n; i++)
        visited[i]=0;

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

    printf("BFS traversal \n");
    BFS(start);
    return 0;
}
```

output: Enter number of vertices: 4
Enter adjacency matrix
0 1 1 0
0 0 1 0
1 0 0 1
0 0 0 0

Date
Page

Enter starting vertex: 2
BFS traversal : 2031

code:

```
#include<stdio.h>
#define MAX 10

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

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

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

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

    for(int i=0;i<n;i++)
        visited[i]=0;
```

```
    printf("DFS Traversal starting from vertex 0:\n");
    DFS(0);
    return 0;
}
```

output: Enter number of vertices: 6
Enter adjacency matrix:
0 1 1 0 0 0
0 0 0 1 1 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 1
0 0 0 0 0 0
DFS Traversal starting from vertex 0:
0 1 3 4 5 2

16/12/25