

4D
action.

ADA LAB Test 1

18M19CS176
VAKAMALLA KEERTHI
PRIYA

4) a) Print all the nodes reachable from a given graph starting node in a digraph using BFS method.

b) Check whether a given graph is connected or not using DFS method.

```
a) #include <stdio.h>
#include <time.h>
int a[20][20], visited[20], n, i, j, f=0, r=-1;

void bfs(int v)
{
    for (i=1; i<=n; i++)
        if (a[v][i] && !visited[i])
            q[++r]=i;
            if (f<=r)
            {
                visited[q[f]]=1;
                bfs(q[f++]);
            }
}

int main()
{
    int v;
    clock_t start, end;
    double t;
    printf("Enter the number of vertices:");
```

```
scanf("%i", &n);
for(i=1; i<=n; i++)
```

```
{
    g[i] = 0;
    visited[i] = 0;
}
```

```
printf("\n Enter graph data in matrix form: \n");
```

```
for(i=1; i<=n; i++)
```

```
{
    for(j=1; j<=n; j++)
```

```
{
    scanf("%i", &g[i][j]);
}
```

```
}
```

```
printf("Enter the starting vertex: ");
```

```
scanf("%i", &v);
```

```
bfs(v);
```

```
printf("\n The nodes which are reachable are:");
```

```
for(i=1; i<=n; i++)
```

```
{
    if(visited[i])
        printf("%i", i);
```

```
else
```

```
{
    printf("\n BFS is not possible. All nodes are not  
reachable!");
```

```
break;
```

```
}
```

```
}
```

```

start = clock();
bfs(v);
end = clock();
t = ((double)(end - start)) / (CLOCKS_PER_SEC);
printf("\n");
printf("\n Time taken by BFS : %.4f\n", t);
printf("\n");
return 0;
}

```

(b) #include <stdio.h>
#include <conio.h>
int a[10][10];
void dfs(int n, int cost[10][10], int u, int s[])
{
int v;
s[u] = 1;
for (v = 0; v < n; v++)
{
if ((cost[u][v] == 1) && (s[v] == 0))
dfs(n, cost, v, s);
}
}
void main()
{
int n, i, j, cost[10][10], s[10], u, flag;
clrscr();
printf("Enter the number of nodes\n");
scanf("%d", &n);
printf("Enter the adjacency matrix\n");
for (i = 0; i < n; i++)
{

```

for (i=0; i<n; i++)
    scanf("%i", &wt[i][i]);
}
con=0;
for (i=0; i<n; i++)
{
    for (j=0; j<n; j++)
    {
        wt[i][j]=0;
        dfs(n, wt, i, 1);
        flag=0;
        for (k=0; k<n; k++)
        {
            if (s[k]==0)
            {
                flag=1;
            }
        }
        if (flag==0)
        {
            con++;
        }
        if (con==1)
            printf("graph is connected\n");
        else
            printf("graph is not connected\n");
        getch();
    }
}

```

Modified

BFS, given an undirected graph, print all connected components line by line.

Number of connected components are 3.

- put one of the graph's vertices at the back of a queue
- Take the front item of the queue & add it to the visited list.
- Add the ones which aren't in the visited list to the back of the queue.
- Repeat it until it gets empty.