

## Experiment Title- 3.1

STUDENT NAME :- RUCHIKA RAJ

UID :- 20BCS9285

SECTION :-615-B

SEMESTER :- 5<sup>TH</sup>

SUBJECT:- DESIGN OF ANALYSIS AND ALGORITHM

**AIM :-** Code and analyze to do a depth-first search (DFS) on an undirected graph. Implementing an application of DFS such as (i) to find the topological sort of a directed acyclic graph, OR (ii) to find a path from source to goal in a maze.

Program Code :-

a) Code and analyze to do a depth-first search (DFS) on an undirected graph

```
b) #include <bits/stdc++.h>
c) using namespace std;
d) class Graph
e) {
f) public:
g)     map<int, bool> visited;
h)     map<int, list<int>> adj;
i)     void addEdge(int v, int w);
j)     void DFS(int v);
k) };
l)
m) void Graph::addEdge(int v, int w)
n)
o) {
p)
q)     adj[v].push_back(w);
r) }
s)
```

```
t) void Graph::DFS(int v)
u)
v) {
w)
x)     visited[v] = true;
y)     cout << v << " ";
z)     list<int>::iterator i;
aa)
bb)     for (i = adj[v].begin(); i != adj[v].end(); ++i)
cc)         if (!visited[*i])
dd)             DFS(*i);
ee) }
ff)
gg) int main()
hh)
ii){
jj)
kk)     Graph g;
ll)     g.addEdge(0, 1);
mm)     g.addEdge(0, 2);
nn)
oo)     g.addEdge(1, 2);
pp)
qq)     g.addEdge(2, 0);
rr)
ss)     g.addEdge(2, 3);
tt)
uu)     g.addEdge(3, 3);
vv)     cout<<"This worksheet belongs to Ruchika Raj (20BCS9285)\n";
ww)
xx)     cout << "Following is Depth First Traversal"
yy)         " (starting from vertex 2) \n";
zz)
aaa)         g.DFS(2);
bbb)
ccc)         return 0;
ddd)     }
eee)
```

## Output :-

```
27     for (i = adj[v].begin(); i != adj[v].end(); ++i)
28         if (!visited[*i])
29             DFS(*i);
30     }
31
32     int main()
33     {
34
35
36         Graph g;
37         g.addEdge(0, 1);
38         g.addEdge(0, 2);
39
40         g.addEdge(1, 2);
41
42         g.addEdge(2, 0);
43
44         g.addEdge(2, 3);
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL AZURE JUPYTER

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\CU\3rd Year\Sem 5\DAA\Worksheet> cd "d:\CU\3rd Year\Sem 5\DAA\Worksheet\" ; if (\$?) { g++ exp3\_1.cpp -o exp3\_1 } ; if (\$?) { .\exp3\_1 }  
This worksheet belongs to Ruchika Raj (20BCS9285)  
Following is Depth First Traversal (starting from vertex 2)  
2 0 1 3  
PS D:\CU\3rd Year\Sem 5\DAA\Worksheet>

fff) to find the topological sort of a directed acyclic graph

## Program Code :

```
#include <bits/stdc++.h>
using namespace std;
class Graph
{
    int V;

    list<int> *adj;

    void topologicalSortUtil(int v, bool visited[],
                            stack<int> &Stack);

public:
    Graph(int V);

    void addEdge(int v, int w);
    void topologicalSort();
};

Graph::Graph(int V)
{
    this->V = V;

    adj = new list<int>[V];
}

void Graph::addEdge(int v, int w)
{
    adj[v].push_back(w);
}

void Graph::topologicalSortUtil(int v, bool visited[],
                                stack<int> &Stack)
{
    visited[v] = true;
```

```
list<int>::iterator i;

for (i = adj[v].begin(); i != adj[v].end(); ++i)
    if (!visited[*i])
        topologicalSortUtil(*i, visited, Stack);

Stack.push(v);
}

void Graph::topologicalSort()
{
    stack<int> Stack;

    bool *visited = new bool[V];
    for (int i = 0; i < V; i++)
        visited[i] = false;

    for (int i = 0; i < V; i++)
        if (visited[i] == false)
            topologicalSortUtil(i, visited, Stack);
    while (Stack.empty() == false)
    {
        cout << Stack.top() << " ";
        Stack.pop();
    }
}

int main()
{
    Graph g(6);

    g.addEdge(5, 2);

    g.addEdge(5, 0);

    g.addEdge(4, 0);

    g.addEdge(4, 1);

    g.addEdge(2, 3);

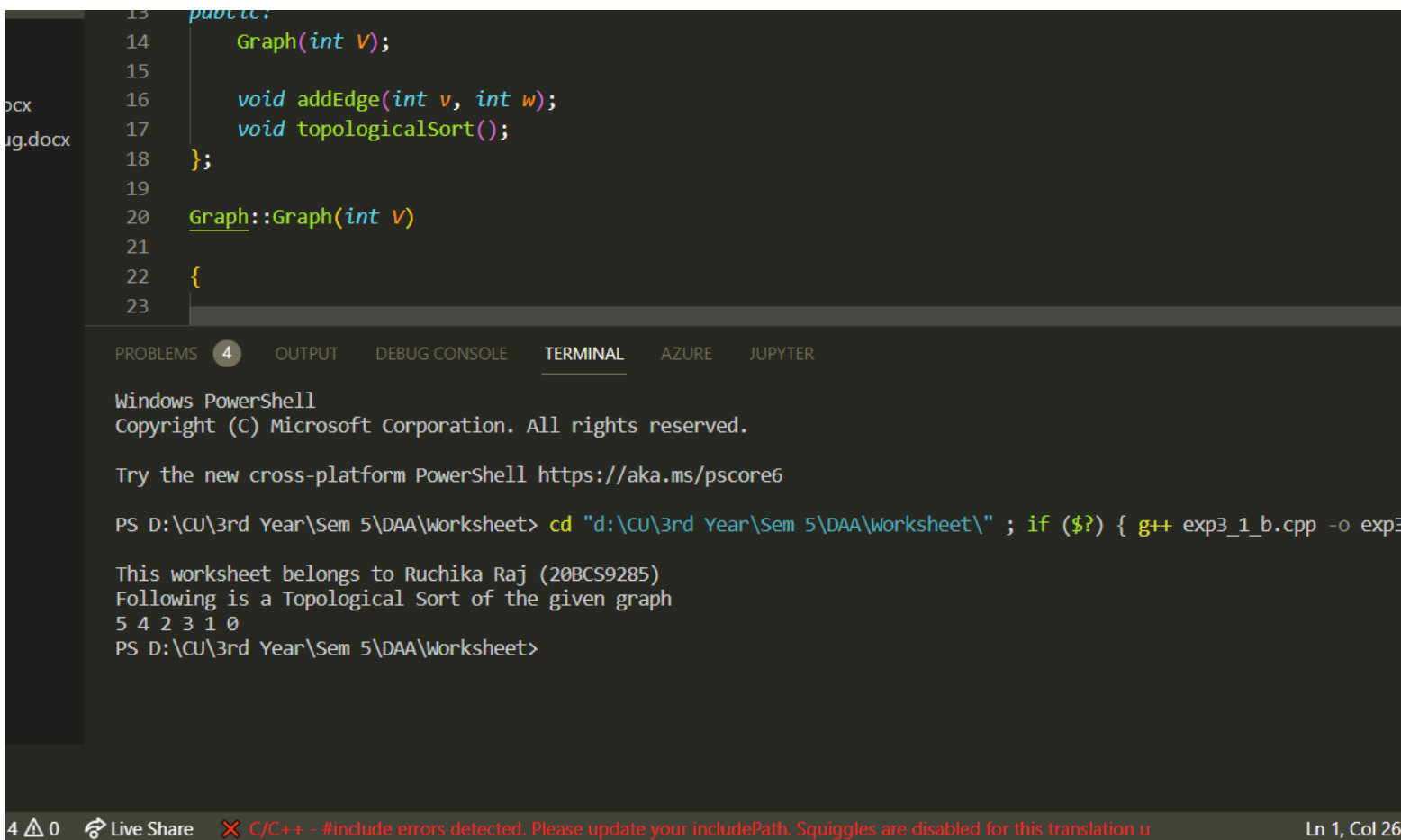
    g.addEdge(3, 1);
```

```
cout << "\nThis worksheet belongs to Ruchika Raj (20BCS9285)\n";

cout << "Following is a Topological Sort of the given "
      "graph \n";
g.topologicalSort();

return 0;
}
```

## Output :-



The screenshot shows a C++ IDE with a code editor on the left and a terminal on the right. The code editor displays the following code:

```
13 public:
14     Graph(int V);
15
16     void addEdge(int v, int w);
17     void topologicalSort();
18 };
19
20 Graph::Graph(int V)
21 {
22 }
23
```

The terminal window shows the output of the program:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\CU\3rd Year\Sem 5\DAA\Worksheet> cd "d:\CU\3rd Year\Sem 5\DAA\Worksheet\" ; if ($?) { g++ exp3_1_b.cpp -o exp3_1_b.exe }

This worksheet belongs to Ruchika Raj (20BCS9285)
Following is a Topological Sort of the given graph
5 4 2 3 1 0
PS D:\CU\3rd Year\Sem 5\DAA\Worksheet>
```

At the bottom of the terminal window, there is a status bar with the following text:

4 ▲ 0 Live Share C/C++ - #include errors detected. Please update your includePath. Squiggles are disabled for this translation u Ln 1, Col 26

C. to find a path from source to goal in a maze.

Program Code :-

```
#include <iostream>
#include <vector>
#include <climits>
#include <cstring>
using namespace std;

bool isSafe(vector<vector<int>> &mat, vector<vector<bool>> &visited, int x, int y)
{
    return (x >= 0 && x < mat.size() && y >= 0 && y < mat[0].size()) && mat[x][y] == 1 &&
!visited[x][y];
}

void findShortestPath(vector<vector<int>> &mat, vector<vector<bool>> &visited, int i, int j, int x,
int y, int &min_dist, int dist)
{
    if (i == x && j == y)
    {
        min_dist = min(dist, min_dist);
        return;
    }

    visited[i][j] = true;

    if (isSafe(mat, visited, i + 1, j))
    {
        findShortestPath(mat, visited, i + 1, j, x, y, min_dist, dist + 1);
    }

    if (isSafe(mat, visited, i, j + 1))
    {
        findShortestPath(mat, visited, i, j + 1, x, y, min_dist, dist + 1);
    }
}
```

```
if (isSafe(mat, visited, i - 1, j))
{
    findShortestPath(mat, visited, i - 1, j, x, y, min_dist, dist + 1);
}

if (isSafe(mat, visited, i, j - 1))
{
    findShortestPath(mat, visited, i, j - 1, x, y, min_dist, dist + 1);
}

visited[i][j] = false;
}

int findShortestPathLength(vector<vector<int>> &mat, pair<int, int> &src, pair<int, int> &dest)
{
    if (mat.size() == 0 || mat[src.first][src.second] == 0 || mat[dest.first][dest.second] == 0)
    {
        return -1;
    }

    int M = mat.size();
    int N = mat[0].size();
    vector<vector<bool>> visited;
    visited.resize(M, vector<bool>(N));
    int min_dist = INT_MAX;
    findShortestPath(mat, visited, src.first, src.second, dest.first, dest.second,
                     min_dist, 0);

    if (min_dist != INT_MAX)
    {
        return min_dist;
    }

    return -1;
}

int main()
{
    vector<vector<int>> mat =
```



```
{

    {1, 1, 1, 1, 1, 0, 0, 1, 1, 1},

    {0, 1, 1, 1, 1, 1, 0, 1, 0, 1},

    {0, 0, 1, 0, 1, 1, 1, 0, 0, 1},

    {1, 0, 1, 1, 1, 0, 1, 1, 0, 1},

    {0, 0, 0, 1, 0, 0, 0, 1, 0, 1},

    {1, 0, 1, 1, 1, 0, 0, 1, 1, 0},

    {0, 0, 0, 0, 1, 0, 0, 1, 0, 1},

    {0, 1, 1, 1, 1, 1, 1, 1, 0, 0},

    {1, 1, 1, 1, 1, 0, 0, 1, 1, 1},

    {0, 0, 1, 0, 0, 1, 1, 0, 0, 1},

};

pair<int, int> src = make_pair(0, 0);
pair<int, int> dest = make_pair(7, 5);

int min_dist = findShortestPathLength(mat, src, dest);

if (min_dist != -1)

{
    cout << "\nThis worksheet belongs to Ruchika Raj (20BCS9285)\n";

    cout << "The shortest path from source to destination "
          "has length "
          << min_dist;
}

else
{

    cout << "Destination cannot be reached from a given source";
}

return 0;
```

}

OUTPUT:

```
118
119     cout << "The shortest path from source to destination "
120         "has length "
121         << min_dist;
122     }
123
124     else
125     {
126
127         cout << "Destination cannot be reached from a given source":
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL AZURE JUPYTER

Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\CU\3rd Year\Sem 5\DAA\Worksheet> cd "d:\CU\3rd Year\Sem 5\DAA\Worksheet\" ; if (\$?) { g++ exp3\_1

This worksheet belongs to Ruchika Raj (20BCS9285)  
The shortest path from source to destination has length 12  
PS D:\CU\3rd Year\Sem 5\DAA\Worksheet>

