

PracticalNo:1

Name:Gunjal Vicky Dnyaneshwar RollNo:39

Title: Implement depth first search algorithm and Breadth First Search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

```
import java.*;

import
java.util.*; class
GraphDFS {

    private int V;

    private LinkedList<Integer> adj[];

    @SuppressWarnings("unchecked")

    GraphDFS(int v)

    {

        V = v;

        adj = new

        LinkedList[v]; for (int i

        = 0; i < v; ++i)

            adj[i] = new LinkedList();

    }

    void addEdge(int v, int w)

    {

        adj[v].add(w);

    }

    void DFSUtil(int v, boolean visited[])

    {

        visited[v] = true;

        System.out.print(v + "

        ");
```

```
Iterator<Integer> i = adj[v].listIterator();
```

```

        while (i.hasNext()) {
            int n =
                i.next(); if
                (!visited[n])
                    DFSUtil(n, visited);
        }
    }

    void DFS(int v)
    {
        boolean visited[] = new
            boolean[V]; DFSUtil(v, visited);
    }

    public static void main(String args[])
    {
        GraphDFS g = new
            GraphDFS(4); g.addEdge(0,
                1);
        g.addEdge(0, 2);
        g.addEdge(1, 2);
        g.addEdge(2, 0);
        g.addEdge(2, 3);
        g.addEdge(3, 3);

        System.out.println(
            "Following is Depth First Traversal "
            + "(starting from vertex
                2)"); g.DFS(2);
    }

```

```
}  
}
```

OUTPUT

```
PS C:\Users\Lenovo> & 'C:\Program Files\Java\jre1.8.0_361\bin\java.exe' '-  
agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:58539'  
'-cp' 'C:\Users\Lenovo\AppData\Local\Temp\vscodesws_8772c\jdt_ws\jdt.ls-java-  
project\bin' 'GraphDFS'
```

Following is Depth First Traversal (starting from

vertex 2) 2 0 1 3

```
PS C:\Users\Lenovo>
```