

**Shri Ramdeobaba College of Engineering & Management Nagpur-13**

**Department of Computer Application**

**Session: 2023-2024**



**Submission for**

**Course Name:** Design Analysis and Algorithm Lab

**Course Code:** MCP546

**Name of the Student:** Jayesh Lalit Nandanwar

**Class Roll No:** 26

**Semester:** MCA II semester

**Shift:** 2

**Batch:** 2

Under the Guidance of

Prof. Manda Ukey

Date of submission: 10/04/2024

## Practical 7

**Aim:** Perform DFS on a directed graph inputted by the user. The starting node is inputted by the user.

Display the order in which DFS is performed on the graph.

Print the time taken to perform this search.

**Code:**

```
import java.util.*;

public class DFS_Practical {
    static class Graph {
        private int V;
        private LinkedList<Integer>[] adj;

        Graph(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) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the number of vertices in the graph: ");
    int V = scanner.nextInt();
    Graph graph = new Graph(V);

    System.out.print("Enter the number of edges in the graph: ");
    int E = scanner.nextInt();
    System.out.println("Enter the edges (source destination):");
    for (int i = 0; i < E; i++) {
        int src = scanner.nextInt();
        int dest = scanner.nextInt();
        graph.addEdge(src, dest);
    }

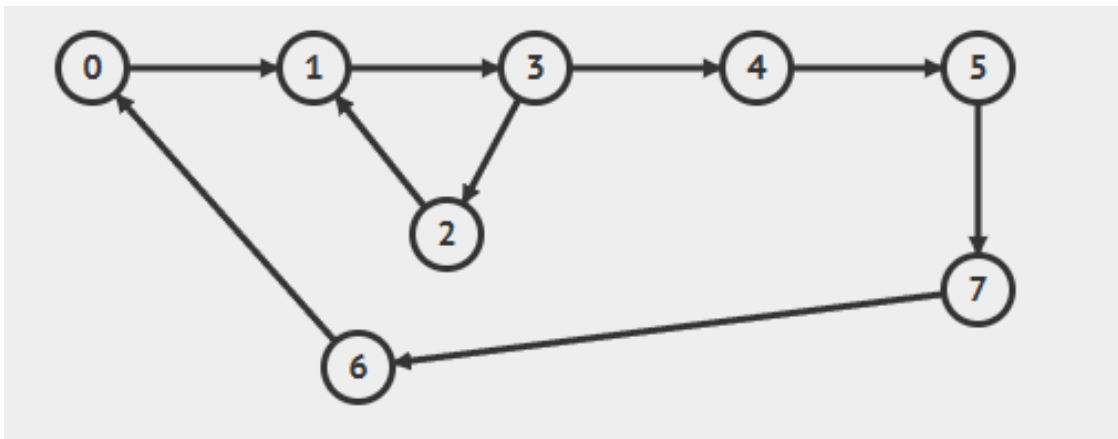
    System.out.print("Enter the starting node for DFS: ");
    int startNode = scanner.nextInt();

    long startTime = System.currentTimeMillis();
    System.out.println("DFS traversal order:");
    graph.DFS(startNode);
    long endTime = System.currentTimeMillis();
    long elapsedTime = endTime - startTime;
    System.out.println("\nTime taken for DFS: " + elapsedTime + " milliseconds");

    scanner.close();
}
}

```

**Graph:**



## Output:

### For Source Node 0:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Work_Files\RCOEM\DAA_Lab\Practical_7> d;; cd 'd:\Work_Files\RCOEM\DAA_Lab\Practical_7'; & 'C:\Program Files\Java\jre-1.8
\bin\java.exe' '-cp' 'C:\Users\MSI\AppData\Roaming\Code\User\workspaceStorage\5de6ff0776a676302c3bdebc15f3910f\redhat.java\jdt_
ws\Practical_7_da3308fa\bin' 'DFS_Practical'
Enter the number of vertices in the graph: 8
Enter the number of edges in the graph: 9
Enter the edges (source destination):
0 1
1 3
3 2
2 1
3 4
4 5
5 7
7 6
6 0
Enter the starting node for DFS: 0
DFS traversal order:
0 1 3 2 4 5 7 6
Time taken for DFS: 1 milliseconds
```

### For Source Node 4:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Work_Files\RCOEM\DAA_Lab\Practical_7> d;; cd 'd:\Work_Files\RCOEM\DAA_Lab\Practical_7'; & 'C:\Program Files\Java\jre-1.8
\bin\java.exe' '-cp' 'C:\Users\MSI\AppData\Roaming\Code\User\workspaceStorage\5de6ff0776abc15f3910f\redhat.java\jbc15f3910f\red
bc15f3910f\redhat.java\jdt_ws\Practical_7_da3308fa\bin' 'DFS_Practical'
Enter the number of vertices in the graph: 8
Enter the number of edges in the graph: 9
Enter the edges (source destination):
0 1
1 3
3 2
2 1
3 4
4 5
5 7
7 6
6 0
Enter the starting node for DFS: 4
DFS traversal order:
4 5 7 6 0 1 3 2
Time taken for DFS: 1 milliseconds
PS D:\Work_Files\RCOEM\DAA_Lab\Practical_7> 
```

**Time Taken: 1 ms**