Pokhara University

Everest Engineering Collage

Sanapa-2, Lalitpur

Lab Report on “SEARCH”

Submitted to
Shailesh Pandey
of
IT Department

Submitted By
Bhupendra Prasad Bhat-[20120010]

Class Roll No :-04

Submitted Date

2023/07/2

DFS.java

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author user23
 */
import java.util.HashMap;
import java.util.LinkedList;

public class DFS {
    public void dfs(Node n, Graph g){
        n.visit();
        System.out.print(n.getName()+",");



        LinkedList <Node> neg = g.getAdjacencyMap().get(n);
        if(neg==null)
            return;
        else{
            for(Node w:g.getAdjacencyMap().get(n)){
                if(!w.isVisited())
                    dfs(w,g);
            }
        }





    }
    public static void main(String[] args){
        Graph g = new Graph(false);

        Node n1 = new Node(1,"Arad");
        Node n2 = new Node(2,"Sibiu");
        Node n3 = new Node(4,"Timisoara");
        //Node n4 = new Node(2,"Zerind");
        //Node n5 = new Node(3,"Oradea");


        g.insertEdge(n1,n2);
        g.insertEdge(n1,n3);
        g.insertEdge(n2,n3);
//        g.insertEdge(n3,n1);



        g.printEdge();

        DFS d = new DFS();
        d.dfs(n1,g);

    }
}



```

Graph.java

```java
import java.util.HashMap;
import java.util.LinkedList;

public class Graph {
    HashMap<Node,LinkedList<Node>> adjacencyMap;
    boolean directed;

    public Graph(boolean dir){
        adjacencyMap = new HashMap<Node,LinkedList<Node>> ();
        directed = dir;
    }
    public void insertEdge(Node Source,Node Desti){
        if(!adjacencyMap.keySet().contains(Source)){
            LinkedList<Node> temp = new LinkedList();
            temp.add(Desti);
            adjacencyMap.put(Source, temp);
        }else{
            LinkedList<Node> temp = adjacencyMap.get(Source);
            temp.add(Desti);
            adjacencyMap.put(Source, temp);
        }
    }

    public HashMap<Node,LinkedList<Node>> getAdjacencyMap(){
        return adjacencyMap;
    }

    public void printEdge(){
        for(Node n: adjacencyMap.keySet()){
            System.out.print(n.getName() + ":");
            for(Node desti:adjacencyMap.get(n)){
                System.out.print(desti.getName()+ ",");
            }
            System.out.print("\n");
        }
    }
}

```

Node.java

```java
public class Node {
    int nodeID;
    String name;
    boolean visited;

    public Node(int id, String city) {
        nodeID = id;
        name = city;
        visited = false;
    }

    public String getName() {
        return name;
    }

    public void visit() {
        visited = true;
    }

    public boolean isVisited() {
        return visited;
    }
}

```