Name – Pranav Salunkhe

TY CS-D

Batch-1

Roll no – 09

Assignment 2

Code –

import java.util.\*;

class Graph{

    private int V;

    private ArrayList<LinkedList<Integer>> adj; // Use an array of LinkedList<Integer>

    Graph(int v){

        V = v;

        adj = new ArrayList<>(v);

        for(int i = 0; i < v; i++)

            adj.add(new LinkedList<>());

    }

    void addEdge(int v, int w){

        adj.get(v).add(w);

        adj.get(w).add(v);

    }

    void BFS(int s){

        boolean[] visited = new boolean[V];

        LinkedList<Integer> queue = new LinkedList<>();

        visited[s] = true;

        queue.add(s);

        while(!queue.isEmpty()){

            s = queue.poll();

            System.out.print(s + " ");

            Iterator<Integer> i = adj.get(s).listIterator();

            while(i.hasNext()){

                int n = i.next();

                if(!visited[n]){

                    visited[n] = true;

                    queue.add(n);

                }

            }

        }

    }

    void DFS(int s){

        boolean[] visited = new boolean[V];

        Stack<Integer> stack = new Stack<>();

        visited[s] = true;

        stack.push(s);

        while(!stack.empty()){

            s = stack.pop();

            System.out.print(s + " ");

            Iterator<Integer> i = adj.get(s).listIterator();

            while(i.hasNext()){

                int n = i.next();

                if(!visited[n]){

                    visited[n] = true;

                    stack.push(n);

                }

            }

        }

    }

}

public class Search {

    public static void main(String[] args){

        Graph g = new Graph(7);

        g.addEdge(0, 1);

        g.addEdge(0, 3);

        g.addEdge(1, 2);

        g.addEdge(1, 3);

        g.addEdge(1, 5);

        g.addEdge(1, 6);

        g.addEdge(2, 5);

        g.addEdge(2, 4);

        g.addEdge(3, 2);

        g.addEdge(3, 4);

        g.addEdge(4, 6);

        System.out.print("BSF - ");

        g.BFS(0);

        System.out.println();

        System.out.print("DSF - ");

        g.DFS(0);

    }

}

Output

