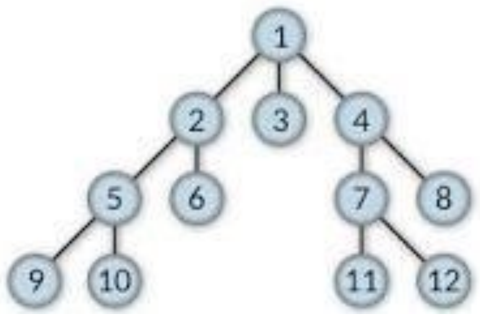
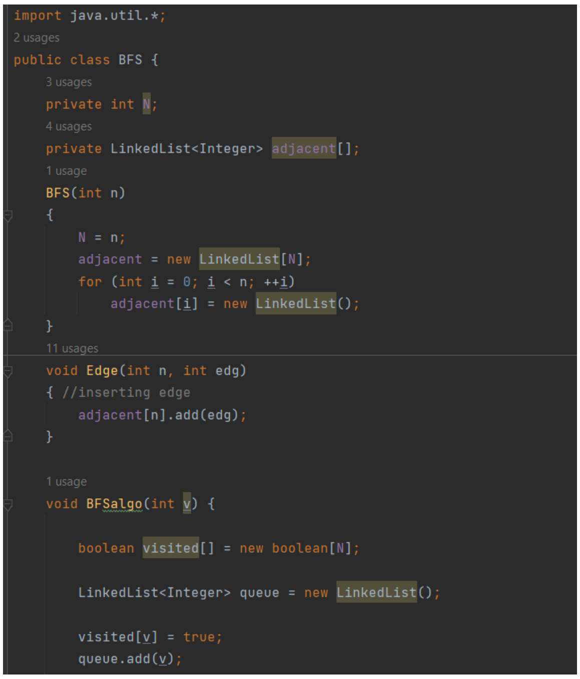
Data Structures and Algorithms

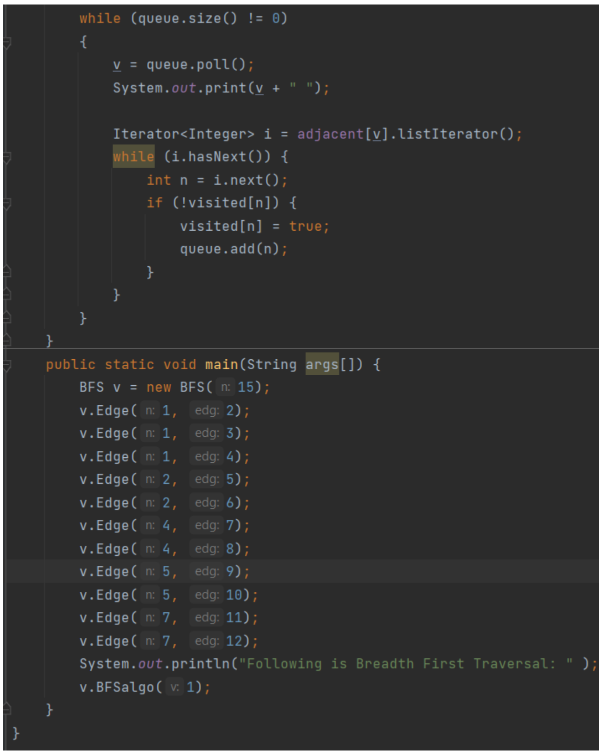
Lab: Graph Algorithm



Code:

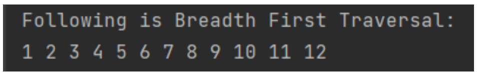


```
import java.util.*;
2 usages
public class BFS {
    3 usages
    private int N;
    4 usages
    private LinkedList<Integer> adjacent[];
    1 usage
    BFS(int n)
    {
        N = n;
        adjacent = new LinkedList[N];
        for (int i = 0; i < n; ++i)
            adjacent[i] = new LinkedList();
    }
    11 usages
    void Edge(int n, int edg)
    { //inserting edge
        adjacent[n].add(edg);
    }

    1 usage
    void BFSalgo(int v) {

        boolean visited[] = new boolean[N];

        LinkedList<Integer> queue = new LinkedList();

        visited[v] = true;
        queue.add(v);
```



```
        while (queue.size() != 0)
        {
            v = queue.poll();
            System.out.print(v + " ");

            Iterator<Integer> i = adjacent[v].listIterator();
            while (i.hasNext()) {
                int n = i.next();
                if (!visited[n]) {
                    visited[n] = true;
                    queue.add(n);
                }
            }
        }
    }
    public static void main(String args[]) {
        BFS v = new BFS( n: 15);
        v.Edge( n: 1,  edg: 2);
        v.Edge( n: 1,  edg: 3);
        v.Edge( n: 1,  edg: 4);
        v.Edge( n: 2,  edg: 5);
        v.Edge( n: 2,  edg: 6);
        v.Edge( n: 4,  edg: 7);
        v.Edge( n: 4,  edg: 8);
        v.Edge( n: 5,  edg: 9);
        v.Edge( n: 5,  edg: 10);
        v.Edge( n: 7,  edg: 11);
        v.Edge( n: 7,  edg: 12);
        System.out.println("Following is Breadth First Traversal: " );
        v.BFSalgo( v: 1);
    }
}
```

Output:



```
Following is Breadth First Traversal:
1 2 3 4 5 6 7 8 9 10 11 12
```

Name: Harshita Pasupuleti

Reg No: 21BCE8421