Write a program to implement BFS Using JAVA

Lab Assignment-9

CSE3002 : Artificial Intelligence

Submitted by:

Jayakumar MHK (18BCE7031)

Under the Guidance of

Prof. Manomita Chakraborty

SCOPE

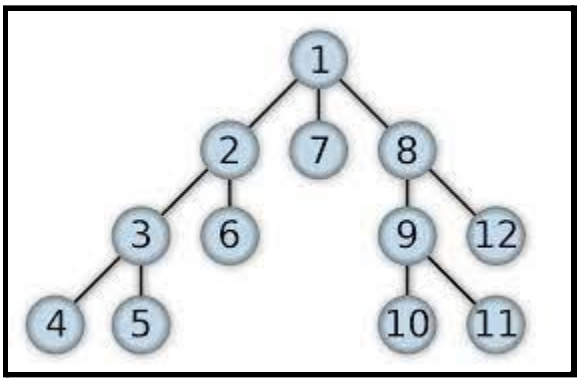
VIT-AP

Task:

Write a program to implement BFS (Breadth-first search)

Solution:

I have used OOPs, ArrayList, and Queue data structure to implement BFS using Java



This is the graph used as a testcase for BFS.

Below is the source code of the same.

```
import java.util.*;
import java.util.LinkedList;
class lab9{
static class Edge{
    int src;
    int nbr;
    public Edge(int s,int d){
        src=s;
        nbr=d;
    }
}
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter number of vertices");
        int v=sc.nextInt();
        ArrayList<Edge>[] graph=new ArrayList[v];
        for(int i=0;i<v;i++){
            graph[i]=new ArrayList<>();
        }
        System.out.println("Enter number of edges");
```

```
        int e=sc.nextInt();
        System.out.println("Enter Each edge ");
        for(int i=0;i<e;i++){
            System.out.print("Edge "+(i+1)+" : ");
            int s=sc.nextInt()-1;
            int d=sc.nextInt()-1;
            graph[s].add(new Edge(s,d));
            graph[d].add(new Edge(d,s));
        }
        boolean []vis=new boolean[v];
        System.out.println("Enter source and goal vertex");
        int src=sc.nextInt()-1;
        int goal=sc.nextInt()-1;
        DFS(src,goal,graph,vis);
    }
    static class Bpair implements Comparable<Bpair>{
        int src;
        int ins;
        public Bpair(int s,int i){
            src=s;
            ins=i;
        }
        public int compareTo(Bpair o){
            return this.ins-o.ins;
        }
    }
    public static void DFS(int src,int goal,ArrayList<Edge> []
graph,boolean [] vis){
        Queue<Integer> qu=new LinkedList<>();
        String asf="";
        qu.add(src);
        while(qu.size()>0){
            int node=qu.remove();

            if(vis[node]==true){
                continue;
            }
            vis[node]=true;

            if(node==goal){
```

```
                System.out.println("BFS Path to reach goal :
"+asf+(goal+1));
            }
            asf+=(node+1)+"->";
            for(Edge e:graph[node]){
                int nbr=e.nbr;
                if(vis[nbr]==false){
                    qu.add(nbr);
                }
            }
        }
    }
}
```

Output below :