CS263 LAB11

Name: Dipean Dasgupta Section:2B

Roll:202151188

Task: Find the Bridges in a given graph.

Solution Code:

```java
import java.util.*;
public class BrdgeGrph {

    public void DFS(int nd, int src, int vist[], int tinst[], int lw[], ArrayList<ArrayList<Integer>> adj, int tm
        vist[nd] = 1;                                   //initializing the src node as visited
        tinst[nd] = lw[nd] = tmcounter++;

        for(Integer it: adj.get(nd)) {                  //adjacent node initiation
            if(it == src)                               //if adj is src, then no need of DFS call
            continue;

            if(vist[it] == 0) {                                 //else call DFS
                DFS(it, nd, vist, tinst, lw, adj, tmcounter);
                lw[nd] = Math.min(lw[nd], lw[it]);

                if(lw[it] > tinst[nd]) {                        //brdge confimation
                    System.out.println(it + "-" +nd);
                }
            } else {
                lw[nd] = Math.min(lw[nd], tinst[it]);
            }
        }
    }
```

```java
  void FindBrge(ArrayList<ArrayList<Integer>> adj, int n)
  {
      int vist[] = new int[n];
      int tinst[] = new int[n];
      int lw[] = new int[n];

      System.out.println(x: "Bridges in the graph are:");

      int tmcounter = 0;
      for(int i = 0;i<n;i++) {
          if(vist[i] == 0) {
              DFS(i, -1, vist, tinst, lw, adj, tmcounter);
          }
      }

  }
```

```java
public static void main(String[] args) {
    int ne = 5;
    ArrayList<ArrayList<Integer> > adj = new ArrayList<ArrayList<Integer> >();
                                                                //creating graph through arraylist
    for (int i = 0; i < ne; i++)
        adj.add(new ArrayList<Integer>());              //Adding sample edges

    adj.get(index: 0).add(e: 1);
    adj.get(index: 1).add(e: 0);

    adj.get(index: 0).add(e: 2);
    adj.get(index: 2).add(e: 0);

    adj.get(index: 1).add(e: 2);
    adj.get(index: 2).add(e: 1);

    adj.get(index: 0).add(e: 3);
    adj.get(index: 3).add(e: 0);

    adj.get(index: 3).add(e: 4);
    adj.get(index: 4).add(e: 3);

    BrdgeGrph obj = new BrdgeGrph();
    obj.FindBrge(adj, ne);                  //finding bridges in the graph
}
```

OUTPUT:

```
D:\Java\ALCS263> d: && cd d:\Java\ALCS263 && cmd /C ""C:\Users\Asus\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.2
8-hotspot\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Asus\AppData\Roaming\Code\User\workspaceS
orage\137c628aba8194fdc47421ded5dd6fa9\redhat.java\jdt_ws\ALCS263_46b310e0\bin BrdgeGrph "
Bridges in the graph are:
4-3
3-0
```

Time complexity:

O(n+m)

Here, N=number of vertices; m=number of edges.