Program on uninformed search methods.(DFS)

```
import java.util.Iterator;
import java.util.LinkedList;
import java.util.List;
public class Main {
 public static void main(String[] args) {
 Graph g = new Graph(8);
 g.AddEdge(1, 2);
 g.AddEdge(1, 5);
 g.AddEdge(2, 3);
 g.AddEdge(2, 5);
 g.AddEdge(3, 4);
 g.AddEdge(4, 5);
 g.AddEdge(4, 6);
 g.AddEdge(5, 6);
 g.DFS();
 }
}
class Graph {
 private int NodeNumber;
 private LinkedList<Integer> AdjacentNodes[];
 Graph(int V) {
 AdjacentNodes = new LinkedList[V];
 for (int i = 0; i < AdjacentNodes.length; i++) {
 AdjacentNodes[i] = new LinkedList();
 }
 NodeNumber = V;
 }
 public void AddEdge(int v, int w) {
 AdjacentNodes[v].add(w);
 }
 void DFSUtil(int v, boolean visited[]) {
 visited[v] = true;
 System.out.println("visiting "+ v + " ");
 Iterator<Integer> i = AdjacentNodes[v].listIterator();
 while (i.hasNext()) {
 int n = i.next();
 if (!visited[n]) {
 DFSUtil(n, visited);
 }
 }
 }
 public void DFS() {
 boolean visited[] = new boolean[NodeNumber];
 for (int i = 0; i < NodeNumber; ++i) {
 if (visited[i] == false) {
 DFSUtil(i, visited);
 }
 }
```

```
 }
}
```

Output:

```
visiting 0
visiting 1
visiting 2
visiting 3
visiting 4
visiting 5
visiting 6
visiting 7
```