```
1
    package graph;
 3
     import java.util.LinkedList;
4
5
    class DFS {
6
7
     public static void DFSUsingAdjacentMatrix (AdjacencyMatrix adjacencyMatrix) {
8
9
     Integer[][] innerAdjacencyMatrix = adjacencyMatrix.getInnerAdjacencyMatrix();
10
11
     boolean[] hasVisited = new boolean[innerAdjacencyMatrix.length];
12
13
            for (int i = 1; i < hasVisited.length; i++) {</pre>
14
                 DFSUsingAdjacentMatrixHelper(i, innerAdjacencyMatrix, hasVisited,
                 adjacencyMatrix);
15
            }
     . . . . .
16
17
18
     private static void DFSUsingAdjacentMatrixHelper(int currentVertexIndex,
19
                                                          Integer[][] innerAdjacencyMatrix,
20
                                                          boolean[] hasVisited,
21
                                                          AdjacencyMatrix
                                                          adjacencyMatrix) {
22
23
     if (hasVisited[currentVertexIndex]) {
24
                 return;
25
26
27
     hasVisited[currentVertexIndex] = true;
28
      System.out.println(adjacencyMatrix.getVertexName(currentVertexIndex));
29
30
31
     for (int j = 0; j < innerAdjacencyMatrix[currentVertexIndex].length; j++) {</pre>
32
33
       if (innerAdjacencyMatrix[currentVertexIndex][j] != null
34
                         && !hasVisited[j]) {
35
36
                    DFSUsingAdjacentMatrixHelper(j, innerAdjacencyMatrix, hasVisited,
                     adjacencyMatrix);
     · · · · · · · · · }
37
38
39
     . . . . . }
40
41
42
      public static void DFSUsingAdjacentTable (AdjacencyTable adjacencyTable) {
43
44
            LinkedList<LinkedList<AdjacencyTable.Node>> innerAdjacencyTable =
            adjacencyTable.getInnerAdjacencyTable();
45
            boolean[] hasVisited = new boolean[innerAdjacencyTable.size()];
46
47
         for (int i = 1; i < innerAdjacencyTable.size(); i++) {</pre>
48
49
                 DFSUsingAdjacentTableHelper(i, innerAdjacencyTable, hasVisited,
                 adjacencyTable);
50
     . . . . . . . . . }
51
     . . . . }
52
53
54
     private static void DFSUsingAdjacentTableHelper(int currentVertexIndex,
55
                                                         LinkedList<LinkedList<AdjacencyTab
                                                         le.Node>> innerAdjacencyTable,
56
                                                         boolean[] hasVisited,
57
                                                         AdjacencyTable adjacencyTable) {
58
59
      if (hasVisited[currentVertexIndex]) {
60
                 return;
61
     62
63
       hasVisited[currentVertexIndex] = true;
64
     System.out.println(innerAdjacencyTable.get(currentVertexIndex).get(0).name);
65
66
     for (int i = 1; i < innerAdjacencyTable.get(currentVertexIndex).size(); i++) {</pre>
```

```
67
 68
        DFSUsingAdjacentTableHelper(
 69
        adjacencyTable.getIndex(
 70
                                             innerAdjacencyTable.get(currentVertexIndex).get(i).name),
 71
                       innerAdjacencyTable, hasVisited, adjacencyTable);
       · · · · · · · · }
 72
 73
 74
 75
 76
       public static void main(String[] args) {
 77
       /*----String[]-vertex-=-{"1",-"2",-"3",-"4",-"5",-"6",-"7"};
 78
 79
 80
        AdjacencyMatrix adjacencyMatrix = new AdjacencyMatrix(vertex);
        AdjacencyMatrix adjacencyMatrix = new AdjacencyMatrix.updateEdge("1", "2", 1);
adjacencyMatrix.updateEdge("3", "1", 6);
adjacencyMatrix.updateEdge("4", "1", 3);
adjacencyMatrix.updateEdge("2", "3", 4);
adjacencyMatrix.updateEdge("2", "4", 4);
adjacencyMatrix.updateEdge("4", "3", 9);
adjacencyMatrix.updateEdge("6", "2", 7);
 81
 82
 83
 84
 85
 86
 87
          adjacencyMatrix.updateEdge("7", "6", 7);
 88
        adjacencyMatrix.updateEdge("7", "4", 7);
 89
          adjacencyMatrix.updateEdge("5", "7", 7);
 90
        adjacencyMatrix.updateEdge("5", "6", 7);
 91
 92
 93
        DFSUsingAdjacentMatrix(adjacencyMatrix); */
 94
 95
        serving[] vertex = {"1", "2", "3", "4", "5", "6", "7"};
 96
 97
        AdjacencyTable adjacencyTable = new AdjacencyTable (vertex);
 98
        adjacencyTable.updateEdge("1", "2", 1);
 99
        adjacencyTable.updateEdge("3", "1", 6);
100
       adjacencyTable.updateEdge("4", "1", 3);
adjacencyTable.updateEdge("2", "3", 4);
adjacencyTable.updateEdge("2", "4", 4);
101
102
103
        adjacencyTable.updateEdge("4", "3", 9);
adjacencyTable.updateEdge("6", "2", 7);
104
105
       adjacencyTable.updateEdge("7", "6", 7);
adjacencyTable.updateEdge("7", "4", 7);
adjacencyTable.updateEdge("5", "7", 7);
adjacencyTable.updateEdge("5", "6", 7);
106
107
108
109
110
111
                DFSUsingAdjacentTable(adjacencyTable);
        . . . . . .
112
113
```