

```

1  package graph;
2
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++) {
14              .... DFSUsingAdjacentMatrixHelper (i, innerAdjacencyMatrix, hasVisited,
15                  .... adjacencyMatrix);
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              .... for (int j = 0; j < innerAdjacencyMatrix[currentVertexIndex].length; j++) {
31                  .... if (innerAdjacencyMatrix[currentVertexIndex][j] != null
32                      .... && !hasVisited[j]) {
33                      .... DFSUsingAdjacentMatrixHelper (j, innerAdjacencyMatrix, hasVisited,
34                          .... adjacencyMatrix);
35                  }
36              }
37          }
38
39      }
40
41
42      .... public static void DFSUsingAdjacentTable (AdjacencyTable adjacencyTable) {
43
44          .... LinkedList<LinkedList<AdjacencyTable.Node>> innerAdjacencyTable =
45              .... adjacencyTable.getInnerAdjacencyTable();
46          .... boolean[] hasVisited = new boolean[innerAdjacencyTable.size()];
47
48          .... for (int i = 1; i < innerAdjacencyTable.size(); i++) {
49              .... DFSUsingAdjacentTableHelper (i, innerAdjacencyTable, hasVisited,
50                  .... adjacencyTable);
51          }
52
53
54          .... private static void DFSUsingAdjacentTableHelper (int currentVertexIndex,
55              ....
56                  .... LinkedList<LinkedList<AdjacencyTab
57                      .... le.Node>> innerAdjacencyTable,
58                      .... boolean[] hasVisited,
59                      .... AdjacencyTable adjacencyTable) {
60
61              .... if (hasVisited[currentVertexIndex]) {
62                  .... return;
63              }
64
65              .... hasVisited[currentVertexIndex] = true;
66              .... System.out.println (innerAdjacencyTable.get (currentVertexIndex).get (0).name);
67
68              .... for (int i = 1; i < innerAdjacencyTable.get (currentVertexIndex).size(); i++) {

```

```

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
78 /* ..... String[] vertex = {"1", "2", "3", "4", "5", "6", "7"};
79
80 ..... AdjacencyMatrix adjacencyMatrix = new AdjacencyMatrix (vertex);
81 ..... adjacencyMatrix.updateEdge ("1", "2", 1);
82 ..... adjacencyMatrix.updateEdge ("3", "1", 6);
83 ..... adjacencyMatrix.updateEdge ("4", "1", 3);
84 ..... adjacencyMatrix.updateEdge ("2", "3", 4);
85 ..... adjacencyMatrix.updateEdge ("2", "4", 4);
86 ..... adjacencyMatrix.updateEdge ("4", "3", 9);
87 ..... adjacencyMatrix.updateEdge ("6", "2", 7);
88 ..... adjacencyMatrix.updateEdge ("7", "6", 7);
89 ..... adjacencyMatrix.updateEdge ("7", "4", 7);
90 ..... adjacencyMatrix.updateEdge ("5", "7", 7);
91 ..... adjacencyMatrix.updateEdge ("5", "6", 7);
92
93 ..... DFSUsingAdjacentMatrix (adjacencyMatrix); */
94
95
96 ..... String[] vertex = {"1", "2", "3", "4", "5", "6", "7"};
97 ..... AdjacencyTable adjacencyTable = new AdjacencyTable (vertex);
98
99 ..... adjacencyTable.updateEdge ("1", "2", 1);
100 ..... adjacencyTable.updateEdge ("3", "1", 6);
101 ..... adjacencyTable.updateEdge ("4", "1", 3);
102 ..... adjacencyTable.updateEdge ("2", "3", 4);
103 ..... adjacencyTable.updateEdge ("2", "4", 4);
104 ..... adjacencyTable.updateEdge ("4", "3", 9);
105 ..... adjacencyTable.updateEdge ("6", "2", 7);
106 ..... adjacencyTable.updateEdge ("7", "6", 7);
107 ..... adjacencyTable.updateEdge ("7", "4", 7);
108 ..... adjacencyTable.updateEdge ("5", "7", 7);
109 ..... adjacencyTable.updateEdge ("5", "6", 7);
110
111 ..... DFSUsingAdjacentTable (adjacencyTable);
112 ..... }
113 }

```