DetectCycle.java

```
1   package com.example.graph;
2
3   import java.util.ArrayList;
4   import java.util.Arrays;
5   import java.util.LinkedList;
6   import java.util.Queue;
7
8   public class DetectCycle {
9       public boolean checkDFS(int node, int parent, boolean vis[], ArrayList<ArrayList<Integer>> adj) {
10          vis[node] = true;
11          for (Integer it : adj.get(node)) {
12 1            if (vis[it] == false) {
13 1                if (checkDFS(it, node, vis, adj) == true)
14 1                    return true;
15 1            } else if (it != parent)
16 1                return true;
17          }
18
19 1        return false;
20      }
21
22      // 0-based indexing Graph
23      public boolean isCycleDFS(int V, ArrayList<ArrayList<Integer>> adj) {
24          boolean vis[] = new boolean[V];
25
26 2        for (int i = 0; i < V; i++) {
27 1            if (vis[i] == false) {
28 1                if (checkDFS(i, -1, vis, adj))
29 1                    return true;
30              }
31          }
32
33 1        return false;
34      }
35
36      static boolean checkBFS(ArrayList<ArrayList<Integer>> adj, int s,
37              boolean vis[], int parent[]) {
38          Queue<Node> q = new LinkedList<>(); // BFS
39          q.add(new Node(s, -1));
40          vis[s] = true;
41
42          // until the queue is empty
43 1        while (!q.isEmpty()) {
44              // source node and its parent node
45              int node = q.peek().first;
46              int par = q.peek().second;
47              q.remove();
48
49              // go to all the adjacent nodes
50              for (Integer it : adj.get(node)) {
51 1                if (vis[it] == false) {
52                      q.add(new Node(it, node));
53                      vis[it] = true;
54                  }
55
56                  // if adjacent node is visited and is not its own parent node
57 1                else if (par != it)
58 1                    return true;
59              }
60          }
61
62 1        return false;
63      }
64
65      // function to detect cycle in an undirected graph
66      public boolean isCycleBFS(int V, ArrayList<ArrayList<Integer>> adj) {
67          boolean vis[] = new boolean[V];
68 1        Arrays.fill(vis, false);
69          int parent[] = new int[V];
70 1        Arrays.fill(parent, -1);
71
```

```
72 2         for (int i = 0; i < V; i++)
73 1             if (vis[i] == false)
74 1                 if (checkBFS(adj, i, vis, parent))
75 1                     return true;
76
77 1         return false;
78       }
79
80   }
81
82   class Node {
83       int first;
84       int second;
85
86       public Node(int first, int second) {
87           this.first = first;
88           this.second = second;
89       }
90   }
```

Mutations

12	1. negated conditional → KILLED
13	1. negated conditional → KILLED
14	1. replaced boolean return with false for com/example/graph/DetectCycle::checkDFS → KILLED
15	1. negated conditional → KILLED
16	1. replaced boolean return with false for com/example/graph/DetectCycle::checkDFS → KILLED
19	1. replaced boolean return with true for com/example/graph/DetectCycle::checkDFS → KILLED
26	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
27	1. negated conditional → KILLED
28	1. negated conditional → KILLED
29	1. replaced boolean return with false for com/example/graph/DetectCycle::isCycleDFS → KILLED
33	1. replaced boolean return with true for com/example/graph/DetectCycle::isCycleDFS → KILLED
43	1. negated conditional → KILLED
51	1. negated conditional → KILLED
57	1. negated conditional → KILLED
58	1. replaced boolean return with false for com/example/graph/DetectCycle::checkBFS → KILLED
62	1. replaced boolean return with true for com/example/graph/DetectCycle::checkBFS → KILLED
68	1. removed call to java/util/Arrays::fill → SURVIVED
70	1. removed call to java/util/Arrays::fill → SURVIVED
72	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
73	1. negated conditional → KILLED
74	1. negated conditional → KILLED
75	1. replaced boolean return with false for com/example/graph/DetectCycle::isCycleBFS → KILLED
77	1. replaced boolean return with true for com/example/graph/DetectCycle::isCycleBFS → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- com.example.graph.DetectCycleTest.testDisconnectedTwoNodeGraph(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testSingleNodeNoEdges(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testEmptyGraph(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testAcyclicThreeNodeGraph(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testCyclicFourNodeGraph1(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testCyclicFourNodeGraph(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testAcyclicThreeNodeGraph1(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testSingleNodeNoEdges1(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testDisconnectedTwoNodeGraph1(com.example.graph.DetectCycleTest) (0 ms)
- com.example.graph.DetectCycleTest.testEmptyGraph1(com.example.graph.DetectCycleTest) (0 ms)

Report generated by PIT 1.15.0