# Bipartite.java

```
   package GraphAlgorithmsTesting;

   import java.util.ArrayList;
   import java.util.Arrays;

   /**
    * Given an adjacency list of a graph adj of V no. of vertices having 0 based
    * index. Check whether the graph is bipartite or not.
    *
    * Input : {{0, 1, 0, 1}, {1, 0, 1, 0}, {0, 1, 0, 1}, {1, 0, 1, 0}}
    *
    * Output : YES
    */
   public class Bipartite {

           private static boolean bipartite(
                           int V,
                           ArrayList<ArrayList<Integer>> adj,
                           int[] color,
                           int node
                           ) {
1                  if (color[node] == -1) {
                           color[node] = 1;
                   }
                   for (Integer it : adj.get(node)) {
1                          if (color[it] == -1) {
1                                  color[it] = 1 - color[node];
1                                  if (bipartite(V, adj, color, it) == false) {
1                                          return false;
                                   }
1                          } else if (color[it] == color[node]) {
1                                  return false;
                           }
                   }
1                  return true;
           }

           public static boolean isBipartite(
                           int V,
                           ArrayList<ArrayList<Integer>> adj
                           ) {
1                  int[] color = new int[V + 1];
1                  Arrays.fill(color, -1);

3                  for (int i = 0; i < V; i++) {
1                          if (color[i] == -1) {
1                                  if (!bipartite(V, adj, color, i)) {
```

```
48 1                                          return false;
49                                    }
50                              }
51                        }
52 1                      return true;
53              }
54  }
```

## Mutations

```
22  1. negated conditional → KILLED
26  1. negated conditional → KILLED
27  1. Replaced integer subtraction with addition → KILLED
28  1. negated conditional → KILLED
29  1. replaced boolean return with true for
    GraphAlgorithmsTesting/Bipartite::bipartite → SURVIVED
31  1. negated conditional → KILLED
32  1. replaced boolean return with true for
    GraphAlgorithmsTesting/Bipartite::bipartite → KILLED
35  1. replaced boolean return with false for
    GraphAlgorithmsTesting/Bipartite::bipartite → KILLED
42  1. Replaced integer addition with subtraction → KILLED
43  1. removed call to java/util/Arrays::fill → KILLED
    1. changed conditional boundary → KILLED
45  2. Changed increment from 1 to -1 → KILLED
    3. negated conditional → KILLED
46  1. negated conditional → KILLED
47  1. negated conditional → KILLED
48  1. replaced boolean return with true for
    GraphAlgorithmsTesting/Bipartite::isBipartite → KILLED
52  1. replaced boolean return with false for
    GraphAlgorithmsTesting/Bipartite::isBipartite → KILLED
```

## Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

## Tests examined

- GraphAlgorithmsTesting.AllGraphTesting.[engine:junit-jupiter]/[class:GraphAlgorithmsTesting.AllGraphTesting]/[method:test()] (26 ms)

Report generated by PIT 1.6.8