

RottingOranges.java

```

1  package Graph;
2
3  import java.util.LinkedList;
4  import java.util.Queue;
5
6  public class RottingOranges {
7      static int arr[][];
8      static int n;
9      static int m;
10     static int ans;
11     static int fresh;
12
13     public static void bfs(Queue<int[]> q){
14
15
16         int dir[][] = {{0,1},{0,-1},{-1,0},{1,0}};
17         while(q.size()!=0){
18             int x = q.size();
19
20             for(int l=0; l<x; l++){
21                 int a[] = q.remove();
22                 int i = a[0];
23                 int j = a[1];
24                 arr[i][j]=2;
25
26                 for(int k=0; k<4; k++){
27                     int newR = i+dir[k][0];
28                     int newC = j+dir[k][1];
29
30                     if(newR<0 || newC<0 || newR>=n || newC>=m || arr[newR][newC]==0 || arr[newR][newC]==2) continue;
31
32                     fresh--;
33                     arr[newR][newC] = 2;
34                     q.add(new int[]{newR,newC});
35                 }
36             }
37             ans++;
38         }
39     }
40
41     public static int orangesRotting(int[][] grid) {
42         arr = grid;
43         n = arr.length;
44         m = arr[0].length;
45
46         ans = 0;
47
48         Queue<int[]> q = new LinkedList<>();
49         fresh=0;
50         int zero=0;
51
52         for(int i=0; i<n; i++){
53             for(int j=0; j<m; j++){
54                 if(arr[i][j]==2) q.add(new int[]{i,j});
55                 else if(arr[i][j]==1) fresh++;
56                 else zero++;
57             }
58         }
59         if(zero==m*n) return 0;
60         bfs(q);
61
62         if(fresh==0) return ans-1;
63         else return -1;
64     }
65 }

```

Mutations

```

17  1. negated conditional → KILLED
20  1. changed conditional boundary → KILLED
    2. negated conditional → TIMED_OUT
26  1. negated conditional → KILLED
    2. changed conditional boundary → KILLED
27  1. Replaced integer addition with subtraction → SURVIVED
28  1. Replaced integer addition with subtraction → SURVIVED
30  1. negated conditional → KILLED
    2. negated conditional → KILLED
    3. changed conditional boundary → KILLED
    4. negated conditional → KILLED
    5. negated conditional → KILLED
    6. negated conditional → KILLED
    7. changed conditional boundary → KILLED
    8. changed conditional boundary → KILLED
    9. negated conditional → KILLED

```

	10. changed conditional boundary → KILLED
32	1. Replaced integer subtraction with addition → KILLED
37	1. Replaced integer addition with subtraction → KILLED
52	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
53	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
54	1. negated conditional → KILLED
55	1. negated conditional → KILLED 2. Replaced integer addition with subtraction → KILLED
56	1. Changed increment from 1 to -1 → KILLED
59	1. negated conditional → KILLED 2. Replaced integer multiplication with division → SURVIVED
60	1. removed call to Graph/RottingOranges::bfs → KILLED
62	1. Replaced integer subtraction with addition → KILLED 2. replaced int return with 0 for Graph/RottingOranges::orangesRotting → KILLED 3. negated conditional → KILLED
63	1. replaced int return with 0 for Graph/RottingOranges::orangesRotting → 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

- Graph.RottingOrangesTest.testOrangesRotting(Graph.RottingOrangesTest) (0 ms)

Report generated by [PIT](#) 1.15.0