MaximalRectangle.java

```
1   package DynamicProgramming;
2   import java.util.*;
3
4   public class MaximalRectangle {
5       public int histogram_area(int[] arr)
6       {
7           int n=arr.length;
8           Stack<Integer> st=new Stack<>();
9           int[] left_small=new int[n];
10          int[] right_small=new int[n];
11
12          //Populating the left small
13 2        for(int i=0;i<n;i++)
14          {
15 3            while(!st.isEmpty() && arr[i]<=arr[st.peek()])
16              {
17                  st.pop();
18              }
19 1            if(st.isEmpty())
20              {
21                  left_small[i]=0;
22              }
23              else
24              {
25 1                left_small[i]=st.peek()+1;
26
27              }
28              st.push(i);
29          }
30          //popping out the stack to clear it.
31 1        while(!st.isEmpty())
32          {
33              st.pop();
34          }
35
36          //Populating the right small
37 3        for(int i=n-1;i>=0;i--)
38          {
39 3            while(!st.isEmpty() && arr[i]<=arr[st.peek()])
40              {
41                  st.pop();
42              }
43 1            if(st.isEmpty())
44              {
45 1                right_small[i]=n-1;
46              }
47              else
48              {
49 1                right_small[i]=st.peek()-1;
50
51              }
52              st.push(i);
53          }
54          int maxi=0;
55 2        for(int i=0;i<n;i++)
```

```
56          {
57 3            int area=(right_small[i]-left_small[i]+1)*arr[i];
58              maxi=Math.max(maxi,area);
59          }
60 1        return maxi;
61
62      }
63
64
65      public int maximalRectangle(char[][] matrix) {
66          int maxi=0;
67          int[] arr=new int[matrix[0].length];
68 2        for(int i=0;i<matrix[0].length;i++)
69          {
70              arr[i]=0;
71          }
72 2        for(int i=0;i<matrix.length;i++)
73          {
74 2            for(int j=0;j<matrix[0].length;j++)
75              {
76 1                if(matrix[i][j]=='1')
77                  {
78 1                    arr[j]++;
79
80                  }
81                  else
82                  {
83                      arr[j]=0;
84                  }
85
86              }
87              maxi=Math.max(maxi,histogram_area(arr));
88          }
89 1        return maxi;
90
91      }
92  }
93
```

Mutations

13	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
15	1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED 3. negated conditional → KILLED
19	1. negated conditional → KILLED
25	1. Replaced integer addition with subtraction → KILLED
31	1. negated conditional → KILLED
37	1. negated conditional → KILLED 2. changed conditional boundary → SURVIVED 3. Replaced integer subtraction with addition → KILLED
39	1. changed conditional boundary → SURVIVED 2. negated conditional → KILLED 3. negated conditional → KILLED
43	1. negated conditional → KILLED
45	1. Replaced integer subtraction with addition → KILLED
49	1. Replaced integer subtraction with addition → KILLED
55	1. negated conditional → KILLED 2. changed conditional boundary → KILLED

57	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED 3. Replaced integer multiplication with division → KILLED
60	1. replaced int return with 0 for DynamicProgramming/MaximalRectangle::histogram_area → KILLED
68	1. negated conditional → SURVIVED 2. changed conditional boundary → KILLED
72	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
74	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
76	1. negated conditional → KILLED
78	1. Replaced integer addition with subtraction → KILLED
89	1. replaced int return with 0 for DynamicProgramming/MaximalRectangle::maximalRectangle → 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

- DynamicProgramming.MaximalRectangleTest.testMaximalRectangle(DynamicProgramming.MaximalRectangleTest) (0 ms)

Report generated by PIT 1.15.0