LengthOfLongestSubstring.java

```
1     package com.example;
2
3
4     import java.util.Arrays;
5     import java.util.HashMap;
6     import java.util.HashSet;
7     import java.util.Map;
8     import java.util.Set;
9
10    public class LengthOfLongestSubstring {
11        //Approach-using Set
12        public static int usingSet(String s) {
13            int n = s.length();
14            int maxLength = 0;
15            Set<Character> charSet = new HashSet<>();
16            int left = 0;
17
18 3          for (int right = 0; right < n; right++) {
19 1              if (!charSet.contains(s.charAt(right))) {
20                    charSet.add(s.charAt(right));
21 2                  maxLength = Math.max(maxLength, right - left + 1);
22                } else {
23 1                  while (charSet.contains(s.charAt(right))) {
24                        charSet.remove(s.charAt(left));
25 1                      left++;
26                    }
27                    charSet.add(s.charAt(right));
28                }
29            }
30
31 1          return maxLength;
32        }
33        //Approach-UnorderedMap
34        public static int usingMap(String s) {
35            int n = s.length();
36            int maxLength = 0;
37            Map<Character, Integer> charMap = new HashMap<>();
38            int left = 0;
39
40 3          for (int right = 0; right < n; right++) {
41 3              if (!charMap.containsKey(s.charAt(right)) || charMap.get(s.charAt(right)) < left) {
42                    charMap.put(s.charAt(right), right);
43 2                  maxLength = Math.max(maxLength, right - left + 1);
44                } else {
45 1                  left = charMap.get(s.charAt(right)) + 1;
46                    charMap.put(s.charAt(right), right);
47                }
48            }
49
50 1          return maxLength;
51        }
52        //Approach-Array
53        public static int usingArray(String s) {
54            int n = s.length();
55            int maxLength = 0;
56            int[] charIndex = new int[128];
57 1          Arrays.fill(charIndex, -1);
58            int left = 0;
59
60 3          for (int right = 0; right < n; right++) {
61 2              if (charIndex[s.charAt(right)] >= left) {
62 1                  left = charIndex[s.charAt(right)] + 1;
63                }
64                charIndex[s.charAt(right)] = right;
65 2              maxLength = Math.max(maxLength, right - left + 1);
66            }
67
68 1          return maxLength;
69        }
70    }
```

Mutations

18	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
19	1. negated conditional → KILLED
21	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED
23	1. negated conditional → KILLED
25	1. Changed increment from 1 to -1 → KILLED
31	1. replaced int return with 0 for com/example/LengthOfLongestSubstring::usingSet → KILLED
40	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
41	1. changed conditional boundary → KILLED 2. negated conditional → KILLED 3. negated conditional → KILLED
43	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED
45	1. Replaced integer addition with subtraction → KILLED
50	1. replaced int return with 0 for com/example/LengthOfLongestSubstring::usingMap → KILLED
57	1. removed call to java/util/Arrays::fill → SURVIVED
60	1. changed conditional boundary → KILLED 2. Changed increment from 1 to -1 → KILLED 3. negated conditional → KILLED
61	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
62	1. Replaced integer addition with subtraction → KILLED
65	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED
68	1. replaced int return with 0 for com/example/LengthOfLongestSubstring::usingArray → 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

- com.example.LengthOfLongestSubstringTest.testUsingMapAdditionalCases(com.example.LengthOfLongestSubstringTest) (1 ms)
- com.example.LengthOfLongestSubstringTest.testUsingArray(com.example.LengthOfLongestSubstringTest) (0 ms)
- com.example.LengthOfLongestSubstringTest.testUsingArrayAdditionalCases(com.example.LengthOfLongestSubstringTest) (1 ms)
- com.example.LengthOfLongestSubstringTest.testUsingSetAdditionalCases(com.example.LengthOfLongestSubstringTest) (1 ms)
- com.example.LengthOfLongestSubstringTest.testEdgeCases(com.example.LengthOfLongestSubstringTest) (1 ms)
- com.example.LengthOfLongestSubstringTest.testUsingMap(com.example.LengthOfLongestSubstringTest) (0 ms)
- com.example.LengthOfLongestSubstringTest.testUsingSet(com.example.LengthOfLongestSubstringTest) (0 ms)

Report generated by PIT 1.5.0