LengthOfLongestSubstring.java

```
   package SlidingWindow;

   import java.util.Arrays;
   import java.util.HashMap;
   import java.util.HashSet;
   import java.util.Map;
   import java.util.Set;

   public class LengthOfLongestSubstring {
       //Approach-using Set
       public static int usingSet(String s) {
           int n = s.length();
           int maxLength = 0;
           Set<Character> charSet = new HashSet<>();
           int left = 0;

2          for (int right = 0; right < n; right++) {
1              if (!charSet.contains(s.charAt(right))) {
                   charSet.add(s.charAt(right));
2                  maxLength = Math.max(maxLength, right - left + 1);
               } else {
1                  while (charSet.contains(s.charAt(right))) {
                       charSet.remove(s.charAt(left));
1                      left++;
                   }
                   charSet.add(s.charAt(right));
               }
           }

1          return maxLength;
       }
       //Approach-UnorderedMap
       public static int usingMap(String s) {
           int n = s.length();
           int maxLength = 0;
           Map<Character, Integer> charMap = new HashMap<>();
           int left = 0;

2          for (int right = 0; right < n; right++) {
3              if (!charMap.containsKey(s.charAt(right)) || charMap.get(s.charAt(right)) < left) {
                   charMap.put(s.charAt(right), right);
2                  maxLength = Math.max(maxLength, right - left + 1);
               } else {
1                  left = charMap.get(s.charAt(right)) + 1;
                   charMap.put(s.charAt(right), right);
               }
           }

1          return maxLength;
       }
       //Approach-Array
       public static int usingArray(String s) {
           int n = s.length();
           int maxLength = 0;
           int[] charIndex = new int[128];
1          Arrays.fill(charIndex, -1);
           int left = 0;

2          for (int right = 0; right < n; right++) {
2              if (charIndex[s.charAt(right)] >= left) {
1                  left = charIndex[s.charAt(right)] + 1;
               }
               charIndex[s.charAt(right)] = right;
2              maxLength = Math.max(maxLength, right - left + 1);
           }

1          return maxLength;
68      }
69  }
```

Mutations

17	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
18	1. negated conditional → KILLED
20	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED
22	1. negated conditional → KILLED
24	1. Changed increment from 1 to -1 → KILLED
30	1. replaced int return with 0 for SlidingWindow/LengthOfLongestSubstring::usingSet → KILLED
39	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
40	1. negated conditional → KILLED 2. negated conditional → KILLED 3. changed conditional boundary → KILLED
42	1. Replaced integer addition with subtraction → KILLED 2. Replaced integer subtraction with addition → KILLED
44	1. Replaced integer addition with subtraction → KILLED
49	1. replaced int return with 0 for SlidingWindow/LengthOfLongestSubstring::usingMap → KILLED
56	1. removed call to java/util/Arrays::fill → SURVIVED
59	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
60	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
61	1. Replaced integer addition with subtraction → KILLED
64	1. Replaced integer subtraction with addition → KILLED 2. Replaced integer addition with subtraction → KILLED
67	1. replaced int return with 0 for SlidingWindow/LengthOfLongestSubstring::usingArray → 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

- SlidingWindow.LengthOfLongestSubstringTest.testUsingSet(SlidingWindow.LengthOfLongestSubstringTest) (0 ms)
- SlidingWindow.LengthOfLongestSubstringTest.testUsingMap(SlidingWindow.LengthOfLongestSubstringTest) (0 ms)
- SlidingWindow.LengthOfLongestSubstringTest.testUsingArray(SlidingWindow.LengthOfLongestSubstringTest) (0 ms)

Report generated by PIT 1.15.0