

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(n^2) time | O(1) space
5     public static String longestPalindromicSubstring(String str) {
6         int[] currentLongest = {0, 1};
7         for (int i = 1; i < str.length(); i++) {
8             int[] odd = getLongestPalindromeFrom(str, i - 1, i + 1);
9             int[] even = getLongestPalindromeFrom(str, i - 1, i);
10            int[] longest = odd[1] - odd[0] > even[1] - even[0] ? odd : even;
11            currentLongest =
12                currentLongest[1] - currentLongest[0] > longest[1] - longest[0]
13                    ? currentLongest
14                    : longest;
15        }
16        return str.substring(currentLongest[0], currentLongest[1]);
17    }
18
19    public static int[] getLongestPalindromeFrom(String str, int leftIdx, int rightIdx) {
20        while (leftIdx >= 0 && rightIdx < str.length()) {
21            if (str.charAt(leftIdx) != str.charAt(rightIdx)) {
22                break;
23            }
24            leftIdx--;
25            rightIdx++;
26        }
27        return new int[] {leftIdx + 1, rightIdx};
28    }
29 }
30
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     public static String longestPalindromicSubstring(String str) {
3         // Write your code here.
4         return "";
5     }
6 }
7
```

Custom Output

Raw Output

Submit Code

**Run or submit code when you're ready.**