

Description | Accepted x Editorial Solutions Submissions
Submit

All Submissions

Accepted 143 / 143 testcases passed

Jeevankumar8074 submitted at Jan 29, 2026 10:13

Runtime

3 ms | Beats 97.01%

Analyze Complexity

Memory

8.35 MB | Beats 37.38%

Code | C

```

1 char* longestPalindrome(char* s) {
2     int len = strlen(s);
3     if(len < 2)
4         return s;
5     int start = 0, maxlen = 1;
6     for(int i=0;i<len;i++){
7         int left = i, right = i;
8         while(left>= 0 && right < len && s[left]==s[right]){
9             if(right - left + 1 > maxlen){
10                 start = left;
11                 maxlen = right - left + 1;
12             }
13             left++;
14             right++;
15         }
16     }
17     left = i;
18     right = i + 1;
19     while(left>=0 && right < len && s[left]==s[right]){
20         if(right - left + 1 > maxlen){
21             start = left;
22             maxlen = right - left + 1;
23         }
24         left++;
25         right++;
26     }
27 }
28
29 s[start + maxlen] = '\0';
30 return s + start;
```

View more

Code

```

1 char* longestPalindrome(char* s) {
2     int len = strlen(s);
3     if(len < 2)
4         return s;
5     int start = 0, maxlen = 1;
6     for(int i=0;i<len;i++){
7         int left = i, right = i;
8         while(left>= 0 && right < len && s[left]==s[right]){
9             if(right - left + 1 > maxlen){
10                 start = left;
11                 maxlen = right - left + 1;
12             }
13             left++;
14             right++;
15         }
16     }
17     left = i;
18     right = i + 1;
19     while(left>=0 && right < len && s[left]==s[right]){
20         if(right - left + 1 > maxlen){
21             start = left;
22             maxlen = right - left + 1;
23         }
24         left++;
25         right++;
26     }
27 }
28
29 s[start + maxlen] = '\0';
30 return s + start;
```

Saved