5. Longest Palindromic Substring ★

Question | Editorial Solution | My Submissions (/problems/longest-palindromic-substring/submissions/)

Total Accepted: **150122** Total Submissions: **622459** Difficulty: **Medium** Contributors: **Admin**

Given a string **s**, find the longest palindromic substring in **s**. You may assume that the maximum length of **s** is 1000.

Example:

```
Input: "babad"

Output: "bab"

Note: "aba" is also a valid answer.
```

Example:

```
Input: "cbbd"

Output: "bb"
```

Hide Company Tags | Amazon (/company/amazon/) | Microsoft (/company/microsoft/) | Bloomberg (/company/bloomberg/)

Hide Tags | String (/tag/string/)

Hide Similar Problems | (H) Shortest Palindrome (/problems/shortest-palindrome/) | (E) Palindrome Permutation (/problems/palindrome-permutation/) | (H) Palindrome Pairs (/problems/palindrome-pairs/)

Have you met this question in a real interview? Yes No

Discuss (https://leetcode.com/discuss/questions/oj/longest-palindromic-substring) | Pick One (/problems/random-one-question/)

Notes

C++ </>

```
class Solution {
public:
    //http://articles.leetcode.com/longest-palindromic-substring-part-i

    //classic DP solution
    string longestPalindrome(string s) {
        int N = s.length();
        vector<vector<int>> dp(N, vector<int>(N, 0));
        int maxlen = 0;
        int maxi = -1;
        for(int i = 0; i < N; ++i) {
            dp[i][i] = 1;
            if(maxlen < 1) {
                maxlen = 1;
                maxi = i;
            }
            if(i != 0 && s[i-1]==s[i]) {
                dp[i-1][i] = 1;
                if(maxlen < 2) {
                    maxlen = 2;
                    maxi = i-1;
                }
            }
        }
        for(int len = 3; len <= N; ++len) {
            for(int i = 0; i + len <= N; ++i) {
                int j = i+len-1;
                if(dp[i+1][j-1] && s[i]==s[j]) {
                    dp[i][j] = 1;
                    if(len > maxlen) {
                        maxlen = len;
                        maxi = i;
                    }
                }
            }
        }
        return s.substr(maxi, maxlen);
    }

    // here is suffix array solution
    // https://discuss.leetcode.com/topic/40846/o-nlogn-suffix-array-solution-clear-explanation




    // manacher's algorithm for O(n) solution
    // http://articles.leetcode.com/longest-palindromic-substring-part-ii/
    /*
    string preprocess(const string& s) {
        stringstream ss;
        ss<<"^#";
        for(char c: s) {
            ss<<c<<"#";
        }
        ss<<"$";
        return ss.str();
    }

    string longestPalindrome(string s) {
        string t = preprocess(s);
        int N = t.size();
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
63
64          int c = 0, mx = 0;
65          vector<int> P(N);
66          for (int i = 1; i < N-1; ++i) {
67             int j = 2*c - i;
68             if(mx-i >= P[j]) P[i] = P[j];
69             else P[i] = mx - i;
70
71             while (t[i+P[i]+1] == t[i-P[i]-1])
72                P[i]++;
73             if(i+P[i] > mx) {
74                 c = i;
75                 mx = i+P[i];
76             }
77          }
78
79          int maxLen = 0;
80          int center = 0;
81          for(int i = 1; i<N-1; ++i) {
82              if (maxLen < P[i]) {
83                  maxLen = P[i];
84                  center = i;
85              }
86          }
87
88          return s.substr((center - maxLen - 1)/2, maxLen);
89      }
90      */
91  };
```

Notes

Custom Testcase Shortcut: Command + enter

Run Code | Submit Solution

Submission Result: Accepted (/submissions/detail/83203272/)

More Details > (/submissions/detail/83203272/)

Next challenges: (H) Valid Number (/problems/valid-number) | (H) Integer to English Words (/problems/integer-to-english-words) | (E) Repeated Substring Pattern (/problems/repeated-substring-pattern)

Share your acceptance!

2 53

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)