Longest Palindromic Substring | Set 2

Given a string, find the l or example, if the given string is “forgeeksskeegfor”, th

Recommended: Please solve it on “*PRACTICE*” first, before moving on to the solution.

We have discussed dynamic programming solution in the previous post. The time complexity of the Dynamic Programming based solution is O(n^2) and it requires O(n^2) extra space. We can find the longest palindrome substring in (n^2) time with O(1) extra space. The idea is to generate all even length and odd length palindromes and keep track of the longest palindrome seen so far.

Step to generate odd length palindrome:
Fix a centre and expand in both directions for longer palindromes.

Step to generate even length palindrome
Fix two centre (low and high) and expand in both directions for longer palindromes.

C++

```
// A O(n^2) time and O(1) space program to
// find the longest palindromic substring
#include <bits/stdc++.h>
using namespace std;
```

```
// A utility function to print a substring str[low..high]
void printSubStr(char* str, int low, int high)
{
    for( int i = low; i <= high; ++i )
        cout << str[i];
}

// This function prints the longest palindrome substring (LPS)
// of str[]. It also returns the length of the longest palindrome
int longestPalSubstr(char *str)
{
    int maxLength = 1; // The result (length of LPS)

    int start = 0;
    int len = strlen(str);

    int low, high;

    // One by one consider every character as center point of
    // even and length palindromes
    for (int i = 1; i < len; ++i)
    {
        // Find the longest even length palindrome
        // with center points as i-1 and i.
        low = i - 1;
        high = i;
        while (low >= 0 && high < len && str[low] == str[high])
        {
            if (high - low + 1 > maxLength)
            {
                start = low;
                maxLength = high - low + 1;
            }
            --low;
            ++high;
        }

        // Find the longest odd length palindrome with center
        // point as i
        low = i - 1;
        high = i + 1;
        while (low >= 0 && high < len && str[low] == str[high])
        {
            if (high - low + 1 > maxLength)
            {
                start = low;
                maxLength = high - low + 1;
            }
            --low;
            ++high;
        }
    }

    cout<<"Longest palindrome substring is: ";
    printSubStr(str, start, start + maxLength - 1);

    return maxLength;
}
```

```
// Driver program to test above functions
int main()
{
    char str[] = "forgeeksskeegfor";
    cout<<"\nLength is: "<<longestPalSubstr(str)<<endl;
    return 0;
}

// This is code is contributed by rathbhupendra
```

C

```
// A O(n^2) time and O(1) space program to find the longest palindromic substrir
#include <stdio.h>
#include <string.h>

// A utility function to print a substring str[low..high]
void printSubStr(char* str, int low, int high)
{
    for( int i = low; i <= high; ++i )
        printf("%c", str[i]);
}

// This function prints the longest palindrome substring (LPS)
// of str[]. It also returns the length of the longest palindrome
int longestPalSubstr(char *str)
{
    int maxLength = 1;  // The result (length of LPS)

    int start = 0;
    int len = strlen(str);

    int low, high;

    // One by one consider every character as center point of
    // even and length palindromes
    for (int i = 1; i < len; ++i)
    {
        // Find the longest even length palindrome with center points
        // as i-1 and i.
        low = i - 1;
        high = i;
        while (low >= 0 && high < len && str[low] == str[high])
        {
            if (high - low + 1 > maxLength)
            {
                start = low;
                maxLength = high - low + 1;
            }
            --low;
            ++high;
        }
```

```
        // Find the longest odd length palindrome with center
        // point as i
        low = i - 1;
        high = i + 1;
        while (low >= 0 && high < len && str[low] == str[high])
        {
            if (high - low + 1 > maxLength)
            {
                start = low;
                maxLength = high - low + 1;
            }
            --low;
            ++high;
        }
    }

    printf("Longest palindrome substring is: ");
    printSubStr(str, start, start + maxLength - 1);

    return maxLength;
}

// Driver program to test above functions
int main()
{
    char str[] = "forgeeksskeegfor";
    printf("\nLength is: %d", longestPalSubstr( str ) );
    return 0;
}
```

Java

```
// Java implementation of O(n^2) time and O(1) space method
// to find the longest palindromic substring
public class LongestPalinSubstring
{
    // A utility function to print a substring str[low..high]
    static void printSubStr(String str, int low, int high) {
        System.out.println(str.substring(low, high + 1));
    }

    // This function prints the longest palindrome substring
    // (LPS) of str[]. It also returns the length of the
    // longest palindrome
    static int longestPalSubstr(String str) {
        int maxLength = 1; // The result (length of LPS)

        int start = 0;
        int len = str.length();

        int low, high;

        // One by one consider every character as center
        // point of even and length palindromes
```

```
        for (int i = 1; i < len; ++i)
        {
            // Find the longest even length palindrome with
            // center points as i-1 and i.
            low = i - 1;
            high = i;
            while (low >= 0 && high < len
                    && str.charAt(low) == str.charAt(high)) {
                if (high - low + 1 > maxLength) {
                    start = low;
                    maxLength = high - low + 1;
                }
                --low;
                ++high;
            }

            // Find the longest odd length palindrome with
            // center point as i
            low = i - 1;
            high = i + 1;
            while (low >= 0 && high < len
                    && str.charAt(low) == str.charAt(high)) {
                if (high - low + 1 > maxLength) {
                    start = low;
                    maxLength = high - low + 1;
                }
                --low;
                ++high;
            }
        }

        System.out.print("Longest palindrome substring is: ");
        printSubStr(str, start, start + maxLength - 1);

        return maxLength;
    }

    // Driver program to test above function
    public static void main(String[] args) {

        String str = "forgeeksskeegfor";
        System.out.println("Length is: " +
                           longestPalSubstr(str));
    }

}
// This code is contributed by Sumit Ghosh
```

Python

```
# A O(n^2) time and O(1) space program to find the
#longest palindromic substring

# This function prints the longest palindrome substring (LPS)
```

```
# of str[]. It also returns the length of the longest palindrome
def longestPalSubstr(string):
    maxLength = 1

    start = 0
    length = len(string)

    low = 0
    high = 0

    # One by one consider every character as center point of
    # even and length palindromes
    for i in xrange(1, length):
        # Find the longest even length palindrome with center
    # points as i-1 and i.
        low = i - 1
        high = i
        while low >= 0 and high < length and string[low] == string[high]:
            if high - low + 1 > maxLength:
                start = low
                maxLength = high - low + 1
            low -= 1
            high += 1

        # Find the longest odd length palindrome with center
        # point as i
        low = i - 1
        high = i + 1
        while low >= 0 and high < length and string[low] == string[high]:
            if high - low + 1 > maxLength:
                start = low
                maxLength = high - low + 1
            low -= 1
            high += 1

    print "Longest palindrome substring is:",
    print string[start:start + maxLength]

    return maxLength

# Driver program to test above functions
string = "forgeeksskeegfor"
print "Length is: " + str(longestPalSubstr(string))

# This code is contributed by BHAVYA JAIN
```

C#

```
// C# implementation of O(n^2) time
// and O(1) space method to find the
// longest palindromic substring
using System;

class GFG
```

```
{
// A utility function to print
// a substring str[low..high]
public static void printSubStr(string str,
                               int low, int high)
{
    Console.WriteLine(str.Substring(low,
                      (high + 1) - low));
}

// This function prints the longest
// palindrome substring (LPS) of str[].
// It also returns the length of the
// longest palindrome
public static int longestPalSubstr(string str)
{
    int maxLength = 1; // The result (length of LPS)

    int start = 0;
    int len = str.Length;

    int low, high;

    // One by one consider every
    // character as center point
    // of even and length palindromes
    for (int i = 1; i < len; ++i)
    {
        // Find the longest even length
        // palindrome with center points
        // as i-1 and i.
        low = i - 1;
        high = i;
        while (low >= 0 && high < len &&
               str[low] == str[high])
        {
            if (high - low + 1 > maxLength)
            {
                start = low;
                maxLength = high - low + 1;
            }
            --low;
            ++high;
        }

        // Find the longest odd length
        // palindrome with center point as i
        low = i - 1;
        high = i + 1;
        while (low >= 0 && high < len &&
               str[low] == str[high])
        {
            if (high - low + 1 > maxLength)
            {
                start = low;
                maxLength = high - low + 1;
            }
            --low;
```

```
            ++high;
        }
    }

    Console.Write("Longest palindrome substring is: ");
    printSubStr(str, start, start + maxLength - 1);

    return maxLength;
}

// Driver Code
public static void Main(string[] args)
{
    string str = "forgeeksskeegfor";
    Console.WriteLine("Length is: " +
                      longestPalSubstr(str));
}
}

// This code is contributed by Shrikant13
```

PHP

```
<?php
// A O(n^2) time and O(1) space program to find the longest palindromic substrir

// A utility function to print a substring str[low..high]
function printSubStr($str, $low, $high)
{
    for( $i = $low; $i <= $high; ++$i )
        echo $str[$i];
}

// This function prints the longest palindrome substring (LPS)
// of str[]. It also returns the length of the longest palindrome
function longestPalSubstr($str)
{
    $maxLength = 1;  // The result (length of LPS)

    $start = 0;
    $len = strlen($str);

    // One by one consider every character as center point of
    // even and length palindromes
    for ($i = 1; $i < $len; ++$i)
    {
        // Find the longest even length palindrome with center points
        // as i-1 and i.
        $low = $i - 1;
        $high = $i;
        while ($low >= 0 && $high < $len && $str[$low] == $str[$high])
        {
            if ($high - $low + 1 > $maxLength)
            {
                $start = $low;
```

```
                $maxLength = $high - $low + 1;
            }
            --$low;
            ++$high;
        }

        // Find the longest odd length palindrome with center
        // point as i
        $low = $i - 1;
        $high = $i + 1;
        while ($low >= 0 && $high < $len && $str[$low] == $str[$high])
        {
            if ($high - $low + 1 > $maxLength)
            {
                $start = $low;
                $maxLength = $high - $low + 1;
            }
            --$low;
            ++$high;
        }
    }

    echo "Longest palindrome substring is: ";
    printSubStr($str, $start, $start + $maxLength - 1);

    return $maxLength;
}

// Driver program to test above functions

$str = "forgeeksskeegfor";
echo "\nLength is: ". longestPalSubstr( $str ) ;
return 0;
?>
```

Output:

```
Longest palindrome substring is: geeksskeeg
Length is: 10
```

Time complexity: O (n^2) where n is the length of input string.
Auxiliary Space: O (1)

We will soon be adding more optimized method as separate post.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

Longest Palindromic Substring using Palindromic Tree | Set 3

Longest Palindromic Substring | Set 1

Longest Non-palindromic substring

Longest palindromic string possible after removal of a substring

Suffix Tree Application 6 - Longest Palindromic Substring

Manacher's Algorithm - Linear Time Longest Palindromic Substring - Part 4

Manacher's Algorithm - Linear Time Longest Palindromic Substring - Part 2

Manacher's Algorithm - Linear Time Longest Palindromic Substring - Part 3

Manacher's Algorithm - Linear Time Longest Palindromic Substring - Part 1

Maximum length palindromic substring such that it starts and ends with given char

Longest Palindromic Subsequence | DP-12

Print Longest Palindromic Subsequence

Length of longest palindromic sub-string : Recursion

Longest palindromic String formed using concatenation of given strings in any order

Longest substring with count of 1s more than 0s

Improved By : shrikanth13, chitranayal, rathbhupendra

Article Tags : Strings Accolite Amazon Groupon Microsoft palindrome

Practice Tags : Accolite Amazon Microsoft Groupon Strings palindrome

3.3

To-do Done

Based on **156** vote(s)

Feedback/ Suggest Improvement | Add Notes | Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Comments **Community** 1 **Login**

Recommend 9 Tweet Share

Sort by Newest

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

Yuri • 7 months ago

Why people keep posting comments with a solution to a problem that has solution described in article? Are you okay?

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Courses
Company-wise
Topic-wise
How to begin?

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos