

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

Difficulty Level : Hard • Last Updated : 11 Feb, 2021

Given a string, find the longest substring which is palindrome.

- if the given string is "forgeeksskeegfor", the output should be "geeksskeeg"
- if the given string is "abaaba", the output should be "abaaba"
- if the given string is "abababa", the output should be "abababa"
- if the given string is "abcbabcbabcba", the output should be "abcbabcbabcba"

We have already discussed Naïve [$O(n^3)$] and quadratic [$O(n^2)$] approaches at [Set 1](#) and [Set 2](#). In this article, we will talk about [Manacher's algorithm](#) which finds Longest Palindromic Substring in linear time.

One way ([Set 2](#)) to find a palindrome is to start from the center of the string and compare characters in both directions one by one. If corresponding characters on both sides (left and right of the center) match, then they will make a palindrome.

Let's consider string "abababa".

Here center of the string is 4th character (with index 3) b. If we match characters in left and right of the center, all characters match and so string "abababa" is a palindrome.



Here center position is not only the actual string character position but it could be the position between two characters also.

Consider string "abaaba" of even length. This string is palindrome around the position between 3rd and 4th characters a and a respectively.



To find Longest Palindromic Substring of a string of length N , one way is take each possible $2*N + 1$ centers (the N character positions, $N-1$ between two character positions and 2 positions at left and right ends), do the character match in both left and right directions at each $2*N+ 1$ centers and keep track of LPS. This approach takes $O(N^2)$ time and that's what we are doing in [Set 2](#).

Let's consider two strings "abababa" and "abaaba" as shown below:



In these two strings, left and right side of the center positions (position 7 in 1st string and position 6 in 2nd string) are symmetric. Why? Because the whole string is palindrome around the center position.

If we need to calculate Longest Palindromic Substring at each $2*N+1$ positions from left to right, then palindrome's symmetric property could help to avoid some of the unnecessary computations (i.e. character comparison). If there is a palindrome of some length L centered at any position P , then we may not need to compare all characters in left and right side at position $P+1$. We already calculated LPS at positions before P and they can help to avoid some of the comparisons after position P .

This use of information from previous positions at a later point of time makes the Manacher's algorithm linear. In [Set 2](#), there is no reuse of previous information and so that is quadratic.

Manacher's algorithm is probably considered complex to understand, so here we will discuss it in as detailed way as we can. Some of it's portions may require multiple reading to understand it properly.

Let's look at string "abababa". In 3rd figure above, 15 center positions are shown. We need to calculate length of longest palindromic string at each of these positions.

- At position 0 there is no LPS at all (no character on left side to compare) so length of LPS will

- At position 2, there is no LPS at all (left and right characters a and b don't match), so length of LPS will be 0.
- At position 3, LPS is aba, so length of LPS will be 3.
- At position 4, there is no LPS at all (left and right characters b and a don't match), so length of LPS will be 0.
- At position 5, LPS is ababa, so length of LPS will be 5.

..... and so on

We store all these palindromic lengths in an array, say L. Then string S and LPS Length L look like below:

Ad
PayPal

注册PayPal企业账户

PayPal
一个账户，收款全球。0费用开户，享卖家保障，赢逾2亿用户。


PayPal

String S		a		b		a		b		a		b		a		
LPS Length L		0	1	0	3	0	5	0	7	0	9	0	11	0	13	



Related Articles

String S		a		b		a		a		b		a		
LPS Length L		0	1	0	3	0	1	6	1	0	3	0	1	0
Position i		0	1	2	3	4	5	6	7	8	9	10	11	12

In LPS Array L:

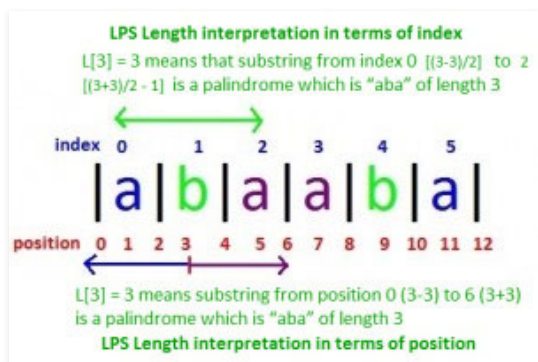
- LPS length value at odd positions (the actual character positions) will be odd and greater than or equal to 1 (1 will come from the center character itself if nothing else matches in left and right side of it)
- LPS length value at even positions (the positions between two characters, extreme left and right positions) will be even and greater than or equal to 0 (0 will come when there is no match in left and right side)

Position and index for the string are two different things here. For a given string S of length N,

LPS length value can be interpreted in two ways, one in terms of index and second in terms of position. LPS value d at position i ($L[i] = d$) tells that:

- Substring from position $i-d$ to $i+d$ is a palindrome of length d (in terms of position)
- Substring from index $(i-d)/2$ to $[(i+d)/2 - 1]$ is a palindrome of length d (in terms of index)

e.g. in string "abaaba", $L[3] = 3$ means substring from position 0 ($3-3$) to 6 ($3+3$) is a palindrome which is "aba" of length 3, it also means that substring from index 0 $[(3-3)/2]$ to 2 $[(3+3)/2 - 1]$ is a palindrome which is "aba" of length 3.



Now the main task is to compute LPS array efficiently. Once this array is computed, LPS of string S will be centered at position with maximum LPS length value.

We will see it in [Part 2](#).

This article is contributed by **Anurag Singh**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Attention reader! Don't stop learning now. Get hold of all the important DSA concepts with the [DSA Self Paced Course](#) at a student-friendly price and become industry ready.

Like 0

[◀ Previous](#)

[Next >](#)

RECOMMENDED ARTICLES

Page : 1 2 3

1 Manacher's Algorithm - Linear Time

2 Z algorithm (Linear time pattern

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

02 Manacher's Algorithm - Linear Time Longest Palindromic Substring - Part 3
17, Dec 14

03 Manacher's Algorithm - Linear Time Longest Palindromic Substring - Part 4
20, Dec 14

04 Longest Palindromic Substring using Palindromic Tree | Set 3
18, Apr 18

06 Minimum length of substring whose rotation generates a palindromic substring
21, May 20

07 Longest Non-palindromic substring
09, May 17

08 Longest Palindromic Substring | Set 2
17, Oct 12

Article Contributed By :



GeeksforGeeks

Vote for difficulty

Current difficulty : [Hard](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [Light_Kiera](#), [vsinghal277](#)

Article Tags : [Microsoft](#) , [palindrome](#) , [Arrays](#) , [Pattern Searching](#) , [Strings](#)

Practice Tags : [Microsoft](#), [Arrays](#), [Strings](#), [palindrome](#), [Pattern Searching](#)

Improve Article

Report Issue

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

16 Comments GeeksforGeeks Disqus' Privacy Policy

Login

Recommend Twitter Facebook Share

Sort by Newest

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !



LOG IN WITH

OR SIGN UP WITH DISQUS

Name



Avatar

Atique Reza • a year ago

Thank you. You explained it pretty well

^ | v • Reply • Share ›



Avatar

Vishal Singla • a year ago

wrong expected output for 4th case i.e.

if the given string is "abcbabcbabcb", the output should be "abcbabcb"
v/s it should be

if the given string is "abcbabcbabcb", the output should be "abcbabcbabcb"

^ | v • Reply • Share ›



Avatar

Kiran Patil • a year ago

<https://www.youtube.com/watch?v=...>

^ | v • Reply • Share ›



Avatar

Rahul Mathur • a year ago

Trying to wrap my head around it since 2 days now, No Luck !

3 ^ | v • Reply • Share ›



Avatar

Mohit Rustagi → Rahul Mathur • a year ago

Try this one. It is simpler I feel.

<https://medium.com/hackernoon/...>

^ | v • Reply • Share ›



Avatar

Abdul-Rahim Taysir → Rahul Mathur • a year ago

I need your help, Please contact me

abdulrahim090a@outlook.com

^ | v 3 • Reply • Share ›



Avatar

Sajal Agrawal • a year ago • edited

How $2*N + 1$ centres are required for consideration?

N: for each odd length Palindromic substring

N - 1: for each even length Palindromic substring

2 end points: the length will always be 0????

^ | v • Reply • Share ›



Avatar

TheDarkKnight • 4 years ago

Someone please share C/C++ code for brute force ie using three loops.

^ | v 2 • Reply • Share ›





Avatar


Qingsong Lv → TheDarkKnight • 2 years ago


Enumerate left end and right end. For each segment, $O(n)$ to judge whether it is palindromic.


^ | v • Reply • Share ›


 **Pratik Bhattacharya** → Ed • a year ago
Avatar Correct.
^ | v • Reply • Share ›

 **Abhishek Madhu** → Ed • 3 years ago
Avatar Yeah, that is what I am thinking about.
1 ^ | v • Reply • Share ›

 **Vinayak Sangar** • 4 years ago
Avatar My code for the solution of this problem
<http://ideone.com/nBQS5d>
^ | v • Reply • Share ›

 **Quinston Pimenta** • 4 years ago
Avatar I made a video explaining this Algorithm with Animtion Graphics on YouTube. Let me know if it helped you! <https://goo.gl/U5igTy>
30 ^ | v 1 • Reply • Share ›

 **Xenial** → Quinston Pimenta • 4 years ago
Avatar Wow, great explanation. My mind has been blown by your explanation. Please post more videos. You just earned another subscriber on your channel :)
1 ^ | v • Reply • Share ›

 **GeeksforGeeks** Mod → Quinston Pimenta • 4 years ago
Avatar Please let us know if you would like to make animation videos for us. Please contact videos@geeksforgeeks.org if you are interested.
4 ^ | v • Reply • Share ›



5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh – 201305

feedback@geeksforgeeks.org

Company

About Us
Careers
Privacy Policy
Contact Us
Copyright Policy

Learn

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

Courses
Company-wise
Topic-wise
How to begin?

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks , Some rights reserved