

Count Palindromic Subsequences

Sunday, November 28, 2021 2:56 PM

Count Palindromic Subsequences

Medium Accuracy: 49.32% Submissions: 51386 Points: 4

Given a string `str` of length `N`, you have to find number of palindromic subsequence (need not necessarily be distinct) which could be formed from the string `str`.

Note: You have to return the answer module 10^9+7 ;

Example 1:

Input:

`Str = "abcd"`

Output:

4

Explanation:

palindromic subsequence are : "a" , "b" , "c" , "d"

Example 2:

Input:

`Str = "aab"`

Output:

4

Explanation:

palindromic subsequence are : "a" , "a" , "b" , "aa"

Your Task:

You don't need to read input or print anything. Your task is to complete the function **countPs()** which takes a string str as input parameter and returns the number of palindromic subsequences.

Expected Time Complexity: $O(N*N)$

Expected Auxiliary Space: $O(N*N)$

Constraints:

$1 \leq \text{length of string str} \leq 1000$

[View Bookmarked Problems](#)

From <<https://practice.geeksforgeeks.org/problems/count-palindromic-subsequences/1>>

Explanation

Dynamic Programming — Longest Palindromic Subsequence

August 4, 2015 by [Sumit Jain](#)

Objective: Given a string, find a longest palindromic subsequence in it.

What is Longest Palindromic Subsequence: A longest palindromic subsequence is a sequence that appears in the same relative order, but not necessarily contiguous(not substring) and palindrome in nature(means the subsequence will read same from the front and back.

Example:

String A = " AABCDEBAZ";

Longest Palindromic subsequence: ABCBA or ABDBA or ABEBA

There are many subsequences can be found which are palindrome like, AA, BCB, ABA, BB etc but we need to find the one with the maximum length.

A	A	B	C	D	E	B	A	Z
---	---	---	---	---	---	---	---	---

Longest Palindromic Subsequence

Approach:

Recursion:

Check every subsequence of in a given string, which means every character as two options, whether it will be included in the subsequence or it will be excluded. If we do it by recursion which means all the sub problems will be solved repeatedly that means **Time Complexity will be $O(2^n)$** .

We can do it better by solving this problem using [Dynamic Programming](#).

Dynamic Programming:

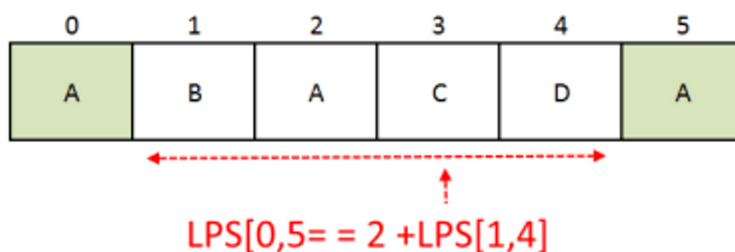
Optimal Substructure:

Given Sequence $A[0....n-1]$

$LPS[0....n-1]$ be the *longest palindromic subsequence* of the given sequence.

Check the first and the last characters of the sequence. Now there are two possibilities, either both the characters same or distinct. We will have to handle both the case.

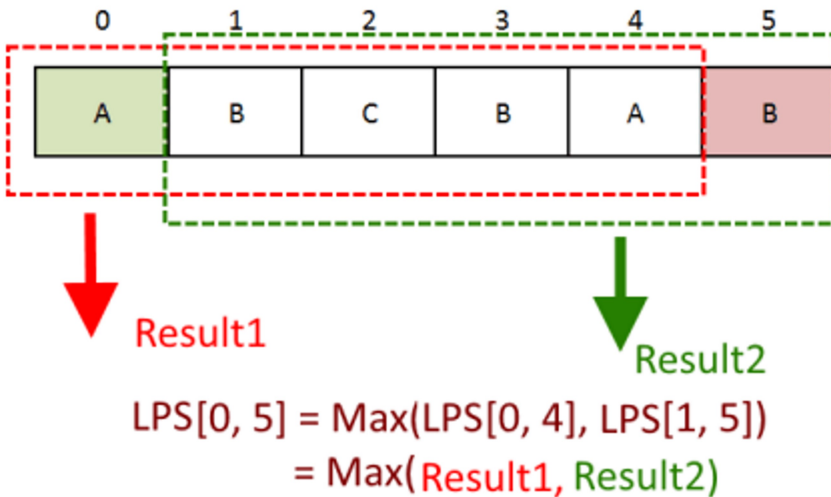
- If both characters are same, add 2 to the result and remove both the characters and solve the problem for the remaining subsequence .



Longest Palindromic Subsequence – 1

- If both characters are different, then solve the problem twice, ignore the first

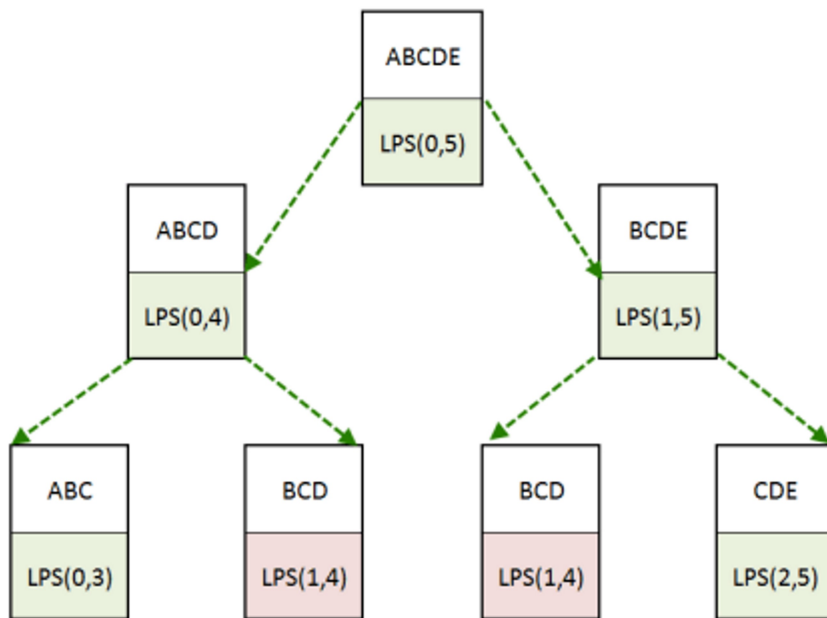
character (keep the last character) and solve the remaining subsequence and then ignore the last character (keep the first character) and solve it and pick the best out of two results.



Overlapping Subproblems:

Say given input string: *ABCDE*

If we solve it recursively, look at the recursion tree, we will be solving the sub-problems repeatedly.



Longest Palindromic Subsequence – Overlapping Subproblems

So while using the dynamic programming, we will solve it in [bottom-up manner](#) and store the results of sub-problems once we solve them for future reference. See the code for the better explanation

Recursive Equations:

$LPS[i, i]$	=	1	Every single character is a palindrome by itself of length 1
$LPS[i, j]$	=	2	if $j=i+1$, sequence has only 2 characters
$LPS[i, j]$	=	$2 + LPS[i-1, j-1]$	If first and last characters are same
$LPS[i, j]$	=	$\text{MAX}(LPS[i+1, j], LPS[i, j-1])$	If first and last characters are not same

Code: [Run This Code](#)

<https://gist.github.com/thmain/b7cb77f0bf8eee557edf48f921c4fd17>

Run This Code

Output:

```

1 2 2 2 2 3 5 5
0 1 1 1 1 3 5 5
0 0 1 1 1 3 3 3
0 0 0 1 1 1 1 1
0 0 0 0 1 1 1 1
0 0 0 0 0 1 1 1
0 0 0 0 0 0 1 1

```

Related Posts:

From <<https://algorithms.tutorialhorizon.com/longest-palindromic-subsequence/>>