

Dynamic

Video-87

Programming



Note :- This playlist is only for explanation of Dns & solutions.

See my "DP Concepts & Dns"
playlist for understanding
DP from scratch...



Leet Code
- 647
Medium

→ 4 Approaches

(Blueprint).

Facebook
Instagram] → code story with MIK

Twitter → cswithMIK



→ code story with MIK

Google

647. Palindromic Substrings

Medium

Topics

Companies

Hint

Given a string `s`, return the number of palindromic substrings in it.

A string is a palindrome when it reads the same backward as forward.

A substring is a contiguous sequence of characters within the string.

Example:- $S = "abc" \rightarrow$

Output = 3 \leftarrow

a✓	abx	abcx
b✓	bcx	
c✓		

$S = "aaa"$

Output = 6 \leftarrow

a✓	aa✓	aaa✓
a✓	a✓	
a✓		
3	2	1

#1.

Brute Force

"Check all substrings"

(i) Substrings

$S = "0000"$

(ii) Check Palindrome

$(0, 2, S)$

0✓			2✓
----	--	--	----

G A A A

r	0	1	2	3	\dots
	a	a	a	a	\dots



bool check (i, j, s) {

$\hat{y} = f(\hat{x})$ (Past solved).

```

    if (s[i] == s[j]) {
        return check(i+1, j-1, s);
    }
    return False;
}

```

#3 Blue Print (Bottom up)

- (*) Count Palindromic Substrings.
- (*) Count odd/even length ^{palind.} Substrings.
- (*) Find Longest Palindromic Substring.
- (*) Find Longest Palindromic Subsequence.
- etc. etc. etc.

$t[i][j]$

State Definition:-

$t[i][j] = \text{True} \rightarrow S[i:j]$ is a Palindrome.
 $\hookrightarrow \text{False} \rightarrow S[i:j]$ is not a Palindrome.

(•) 1 length \rightarrow i \Rightarrow True
 $S = "$ a a a $"$
 $(0,0)$ $(1,1)$ $(2,2)$

Note

$$t[i][i] = \text{True}$$

	0	1	2	3
0	T			
1		T		
2			T	
3				T

• 2 length :-

i j

$$S[i] == S[j]$$

$\underbrace{i+1 == j}_{\text{2 length str substring}} \rightarrow$

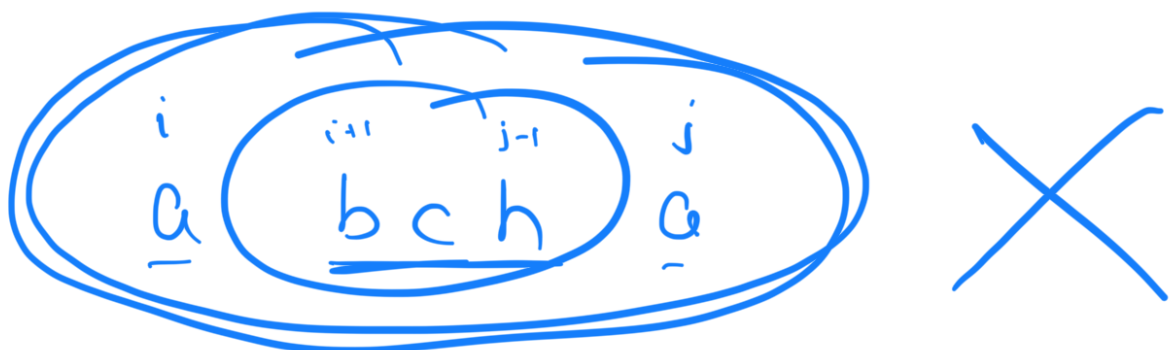
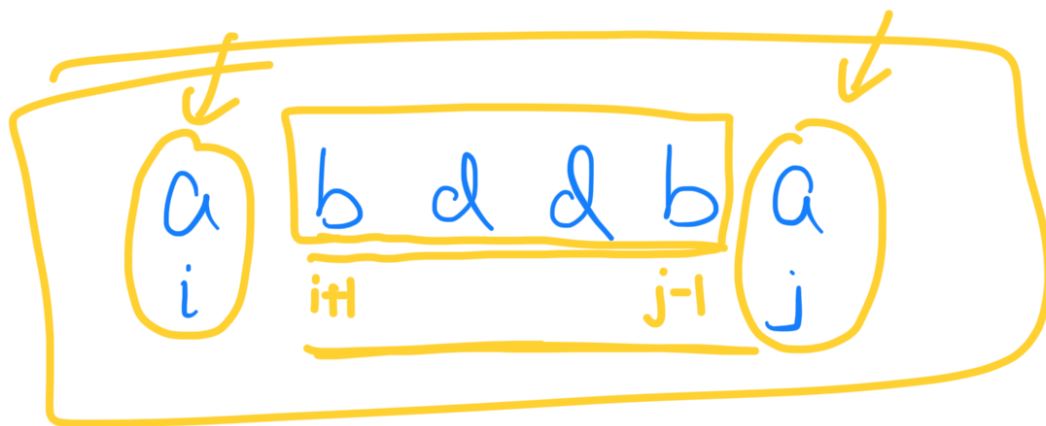
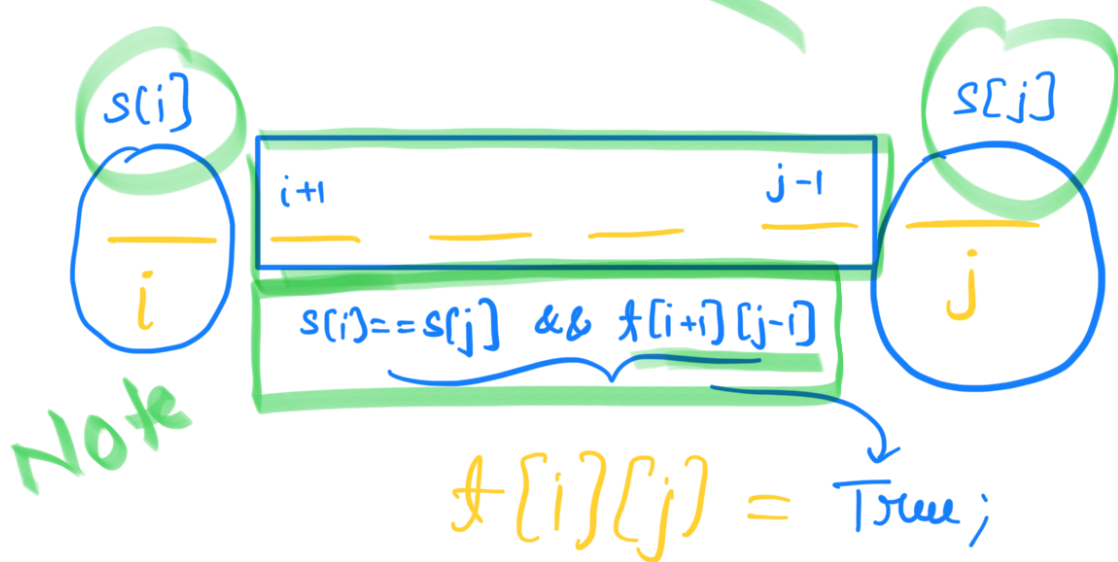
$$i+1 == j$$

Note

$$S[i] == S[j]$$

$i = 3$

- > 2 length (Generic) 



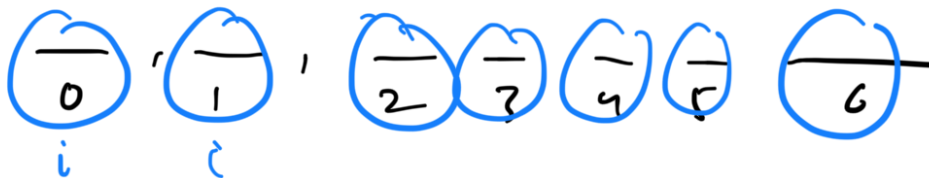
$f[i][i] = s[i] == s[j] \ \&\& \ f[i+1][j-1] = \text{False}$

Now comes the beautiful Part:-



$\{ [] [] \} = \text{False}$

Count = 0 ;



$L=1$ (i, i) $(0, 0)$
 $\Rightarrow L=2$ $(0, 1)$
 $\Rightarrow L=3$ $(0, 2)$

$\{ (0) (0) \} = T$

$\{ (1) (1) \} = T$

for ($L=1$; $L \leq n$; $L++$) {

for ($i=0$; $i+L-1 < n$; $i++$)

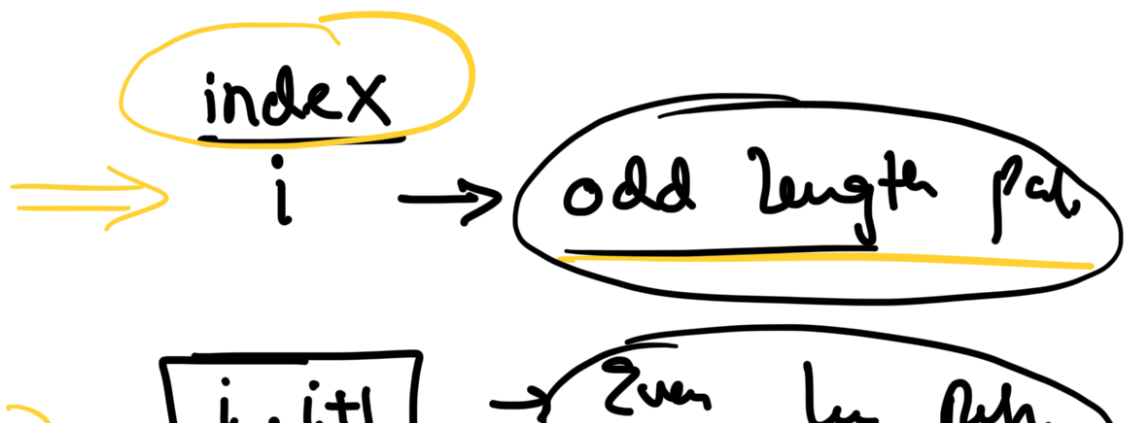
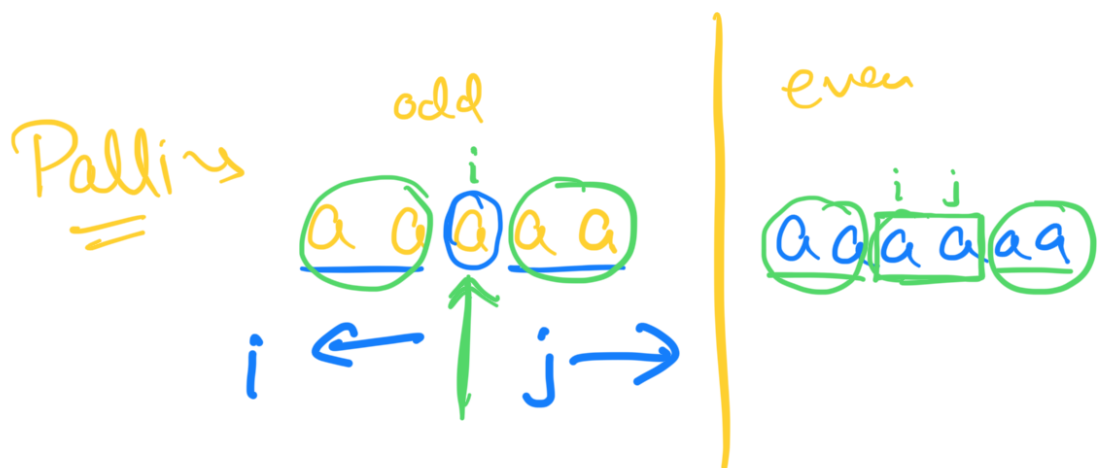
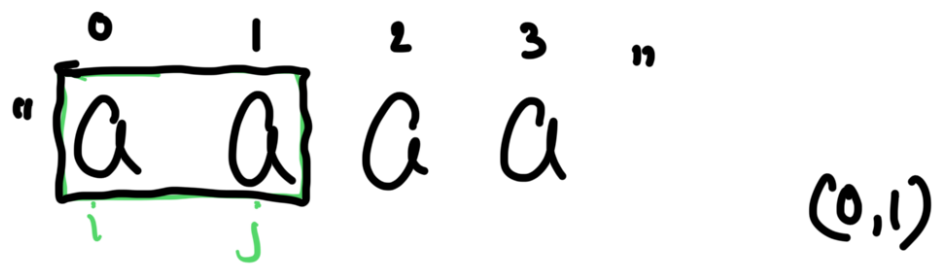
$j = i + L - 1$;



valid index

#4

Smart Approach



for (i = 0 ; i < n ; i++) {

Check (^{center} $\boxed{i, i}$, s, n); // odd & len.

Check ($\boxed{i, i+1}$, s, n); // even len.

}

check (i, j, s, n) {

← i j →

while (i >= 0 && j < n && s[i] == s[j])

{

Count++;

← i--;

j++; →

}

}

Count = 1 + 1

" a a aⁱ "

"C"

check(i, i)
check(i, i+1)

check(i, j)

{ s[i] == s[j].
count++
i--
j++

check(i, j)

s[i] == s[j].
count++
i--
j++

}