

Strings

Strings : → Group of chars
 ↳ sequence of chars
 ↳ Arrays/List of chars

"dabc"
 ↓↓
 abcd

Computer → O/I

ASCII ⇒ American Standard Code of Information Interchange.

'A'	→	65	'a'	→	97	'0'	→	48
'B'	→	66	'b'	→	98	'1'	→	49
:	:	:	:	:	:	:	:	:
'Z'	→	90	'z'	→	122	'q'	→	58
						"10"	→	48

Immutable strings

(Java/python & other language)

String $s_1 = "abc"$

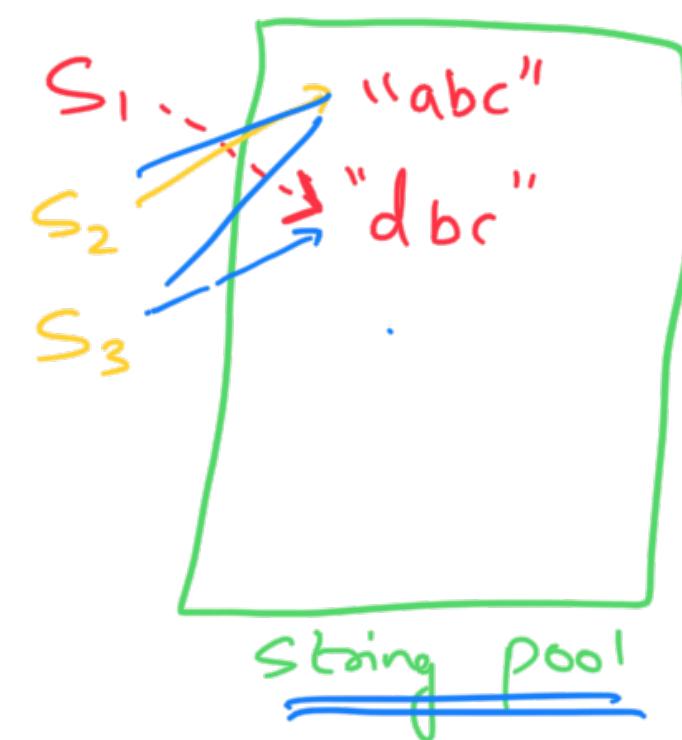
// $s_1[0] = 'd'$

// $s_1 \Rightarrow "dbc"$

String $s_2 = "abc"$

String $s_3 = "abc"$

↳ $s_3 = "dbc"$



Append a char to immutable string

→ $s_1 = "a";$

"a"]

String

$$S_1 = S_1 + \underline{b} \Rightarrow \cancel{O(1)} O(N)$$
$$S_1 = S_1 + \underline{c} \Rightarrow \cancel{O(1)} O(N)$$
$$S_1 = S_1 + \underline{d} \Rightarrow \cancel{O(1)} O(N)$$

If we append N times, $\times e$

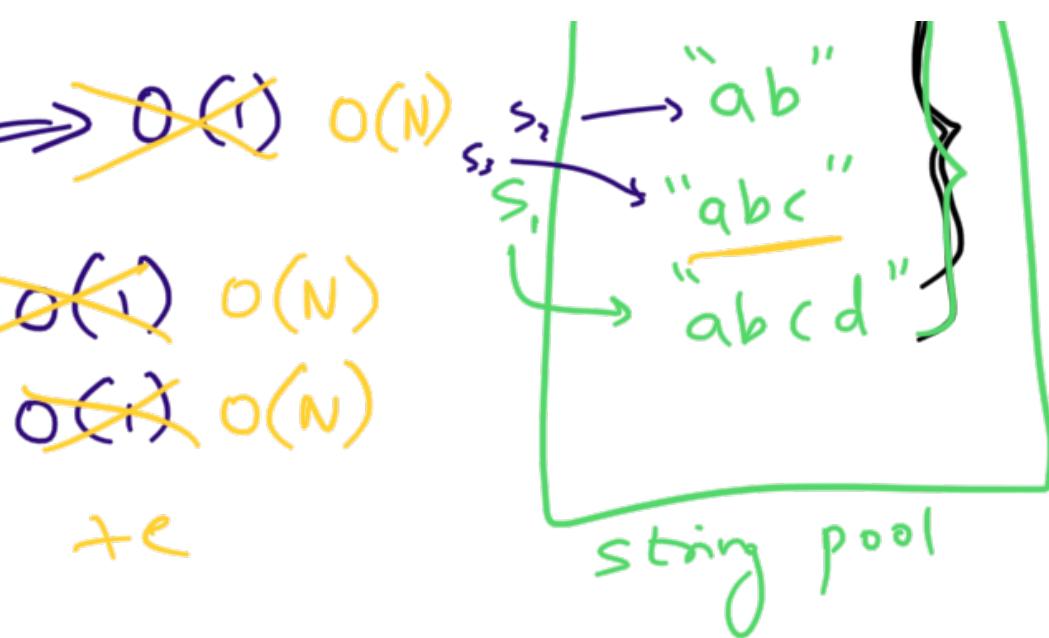
$$T.C \Rightarrow \underline{\underline{O(N^2)}}$$

$$S.C \Rightarrow \underline{\underline{O(N^2)}}$$

$$S_2 = ab$$

$$S_3 = abc$$

$$S_n = ..$$



Garbage Collector cleans out the useless string from memory.

String Builder

StringBuilder

Java

```
StringBuilder sb = new StringBuilder();  
sb.append('b')
```

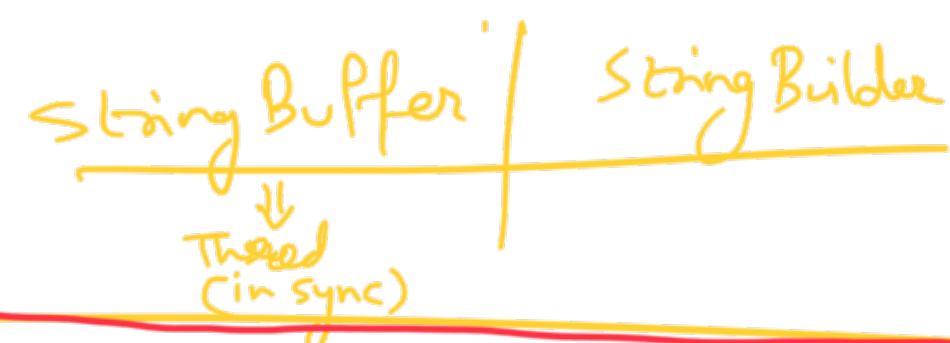
C#

sb.append("any")
sb.charAt(i)

sb.toString()

Java String.

C++ → by default has mutable



char s = 'a'
↓
s = 'a'+1 ⇒ "98" ⇒ 'b'
↓
97

char s = 'b' - 'a' ⇒ (int) = ①

('c' + 'b' - 'a') ⇒ 'c' + ① ⇒ 'd'

Q.1

Given a string s . Toggle the case of every character. (only contain alphabets)

uppercase \rightarrow lowercase

lowercase \rightarrow uppercase

$s : aBc AEd \rightarrow AbCaeD$

[cannot use inbuild method]

```
toggle (s){  
    for (i=0 ; i < s.size() ; i++)  
    {  
        //if s[i] is lower case  
        if (s[i] >='a' && s[i] <='z')  
            s[i] = s[i] ^ 32; //  $s[i] = s[i] - 32$ ;  $s[i] = abs(a - A)$   
        // else s[i] is uppercase.  
        else if (s[i] >='A' && s[i] <='Z')  
            s[i] = s[i] + 32;
```

$S \cup J - \rightarrow L \cup J \rightarrow C$

3

3

'a' → 97
'b' → 98
.....
'z' → 127

$\sim z \rightarrow 122$ → 5th bit is always set

$$\begin{array}{r} \textcircled{1} \\ - 654 \\ \hline 3210 \end{array}$$

'A'	→	65
'B'	→	66
.		
'Z'	→	90

$Z \rightarrow 10$ bit is ~~an~~ 5m 7 6 5 4 3 2

'z' → 122

$$\begin{array}{r} \underline{7} & \underline{6} & \underline{5} & \underline{4} & \underline{3} & \underline{2} & \underline{1} & \underline{0} \\ & | & @ & | & | & | & | & | \\ & 64 & & & & & & & \end{array}$$

s always unsat

7 6 5 3 2 1 0

4 3 2 1 0

0 1 1 0 1 0 1 0

$\text{'Z'} \rightarrow 90$ \downarrow $\underline{\text{o } 1 \text{ 0 } 1 \text{ 1 } 1 \text{ 0 } 1 \text{ 0}}$

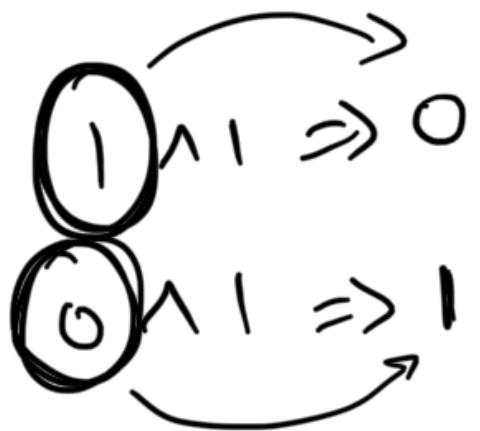
32

$\text{'z'} - \text{'Z'} = 32$

$s[i] = s[i] \wedge (1 \ll 5)$



$\text{'Z'} \Rightarrow 01011010$



$97 \leftarrow \text{'a'} \Rightarrow$

$65 \leftarrow \text{'A'} \Rightarrow$

$\begin{array}{r} 64 \\ 32 \\ \hline 01010000 \\ 01010000 \\ \hline 00000000 \end{array}$

T.C $\Rightarrow O(N)$

S.C $\Rightarrow O(1)$

\rightarrow string builder

Unicode

Devnagri

Q.2 Given a string of lowercase char. Sort it in dict order.

s: dabaed b

ans aabbdd e

Use library sorting function $\rightarrow O(N \log N)$

lowercase alphabet $\rightarrow [a-z] \Rightarrow 26$ unique chars.

d a b a e d b

d : 2

a : 2

b : 2

e : 1

for(a \rightarrow z)

aabbdd e

	0	1	2	25
	2	2	0	2	1	..	
a	a	b	c	-	-	-	z

Hashmap/dict

int count[26] = {0}

char	index
a → 97	0
b → 98	1
c → 99	2
:	:
z → 122	25

s[i] - 'a'
↓
97

// count freq

```
int count[26] = {0};  
for (i=0; i<N; i++)  
{  // index for s[i] ⇒ s[i]-'a'  
    count[s[i]-'a']++;  
}
```

O(1) ⇒ independent of input

// Rebuild the string using count-array in dict order.

```

K = 0;
for (i=0 ; i < 26 ; i++)
{
    // count[i] stores freq. of i+'a'
    Count[i] times {
        for (j=0 ; j < count[i] ; j++)
            S[K] = char(i+'a')
            K++;
    }
}

```

Counting Sort

S.C : $O(1)$ (if string is mutable)
 (extra)

T.C : $O(N) + O(N)$
 (count freq) (Build the string)

T.C : $\underline{O(N)}$

i	j	#iterations
0	$[0 \rightarrow a]$	$ a $
1	$[0 \rightarrow b]$	$ b $
2	$[0 \rightarrow c]$	$ c $
:		:
25	$[0 \rightarrow z]$	$ z $
		size of string O
		$= N$

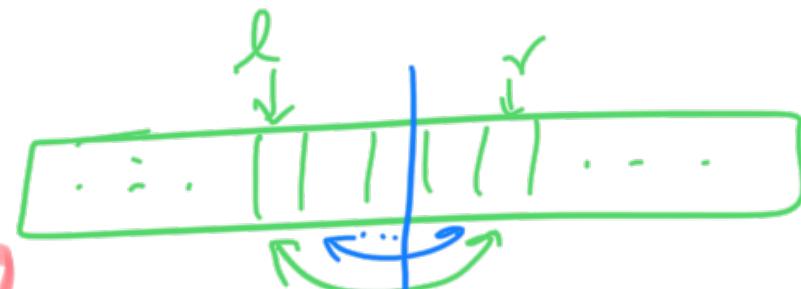
Q. Given a string S & two indices l ' & γ .
 Reverse the substring from l to γ .

0 1 2 3 4 5 6

a b c d e f g

$$l = 2, \gamma = 4$$

ans \Rightarrow a b e d c f g



```
reverse(s, l, γ)
{
    while (l < γ)
        swap(s[l], s[γ])
        l++; γ--;
}
```

T.C $\Rightarrow O(N)$
 S.C $\Rightarrow O(1)$ } mutable string.

Amazon

Q. Given a character array storing a sentence.

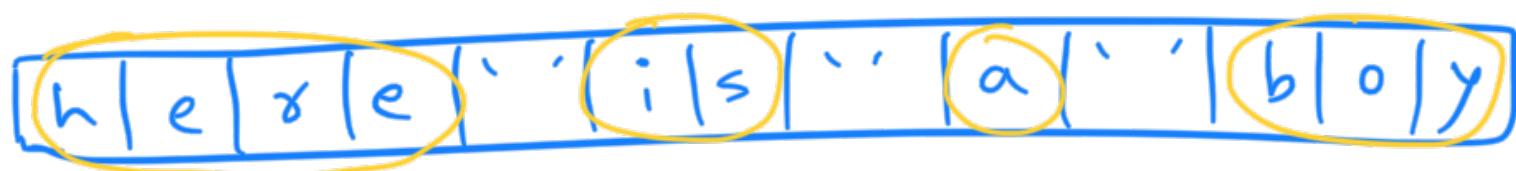
Reverse it word-by-word.

* No extra space allowed

* Every word is separated by a single white space (' ')

* Cannot use .split()

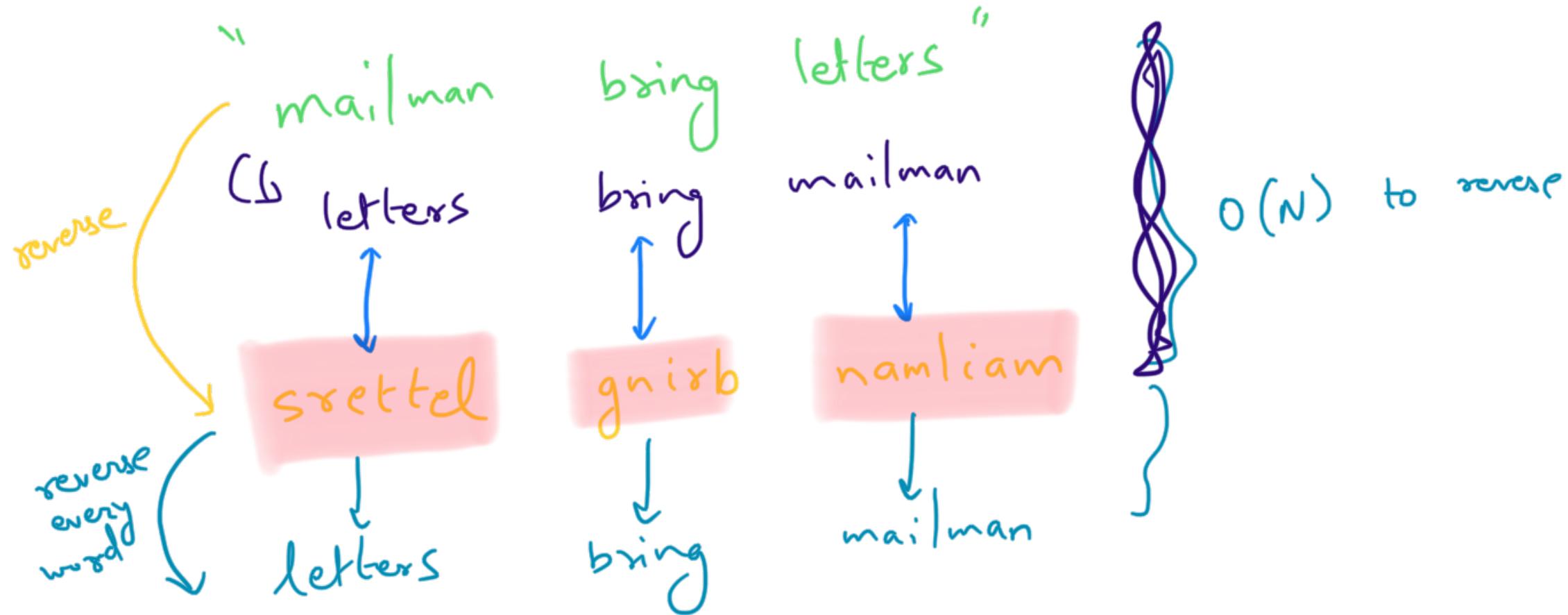
here - is - a - boy



ans ↴ boy a is here

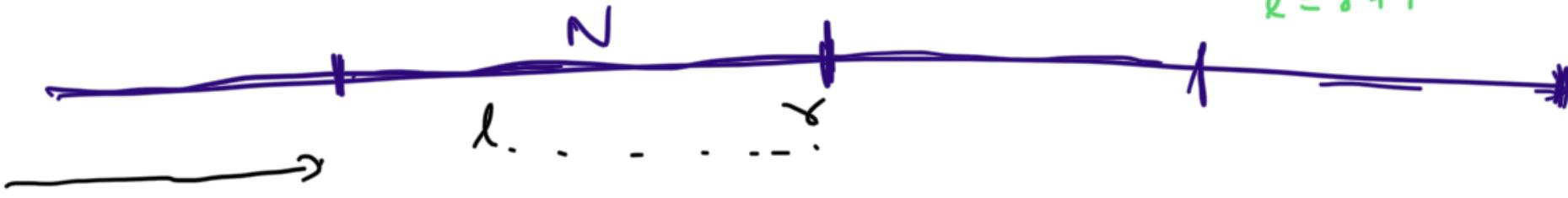
"Are you as clever as I am"

"I am as clever as you Are"



$T.C \Rightarrow O(N) + O(N)$
 (reverse entire array)
 (reverse every word)

l r
 srettel Ognirb Oramliam
 (l → r - 1) (l → r - 1)
 while ($r < N \text{ & } s[r] \neq '$)
 reverse (l, r - 1)
 r = r + 1



$S :$

<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u>
----------	----------	----------	----------

#iter \Rightarrow $\frac{a}{2} + \frac{b}{2} + \frac{c}{2} + \frac{d}{2}$

$$N + \frac{(a+b+c+d)}{2} = N + \binom{N}{2} \Rightarrow O(N)$$

~~Pseudo Code :~~

- ① Reverse entire string \Rightarrow reverse(s, 0, N-1)
- ② iterate using loop
 (\hookrightarrow call reverse on every word).

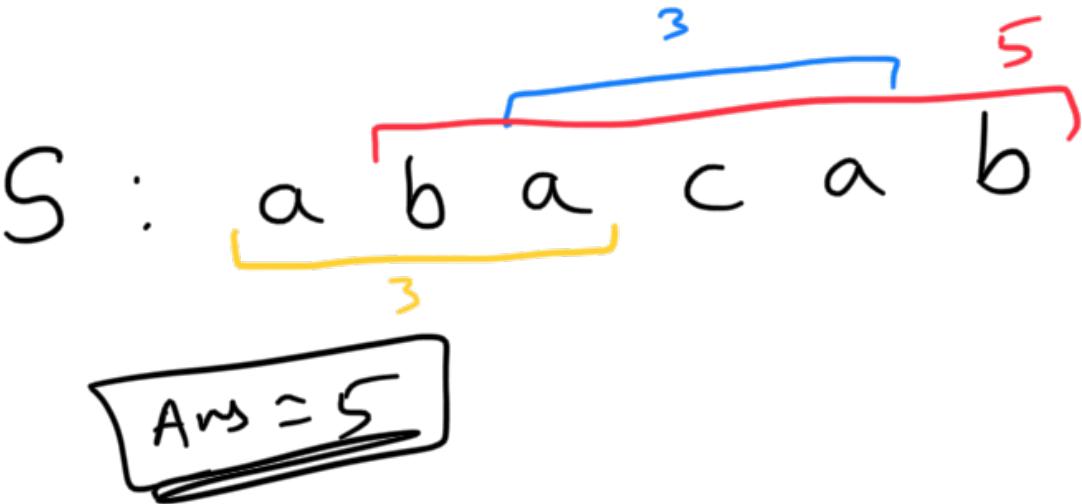
H.W.
 Argon
 Diredi Q.
 Ola

Given a string of size N . (lowercase alphabets)
 Return the length of longest palindromic substring.

n.n. Name : mom

Kalinaoo

nitin
 malayalam $\xrightarrow{\text{reverse}}$ malayalam
 racecar

S : 

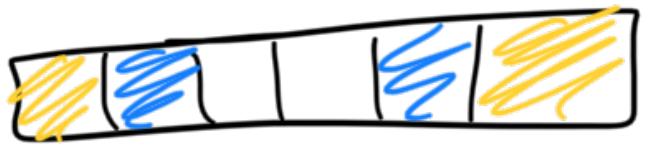
aba $\rightarrow 3$
 aca $\rightarrow 3$
 c $\rightarrow 1$
 b $\rightarrow 1$
 bacab $\rightarrow 5$

S : abca l m

Ans : 1

Brute force
Approach 1

- ① Iterate on all substring : $O(N^2)$
- ② For every substring, check if its pal : $O(N)$
- ③ Get the max length.

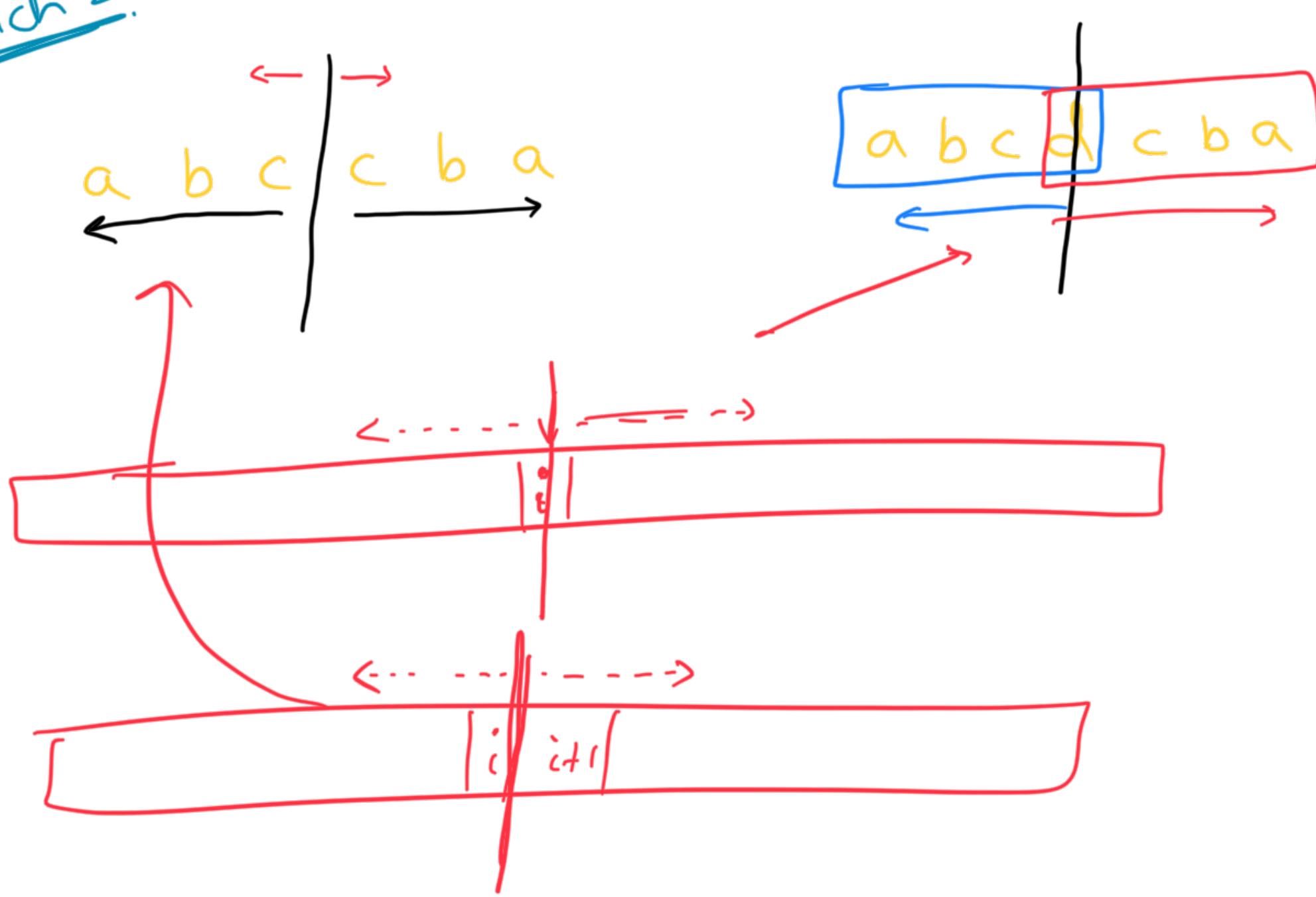


$T.C \Rightarrow O(N * N)$

↓
no. of
substring

↓
check that
substring.

Approach 2:



O ab c | c b a | a
ab c c b a .

→ a b a c a b
a ↑
aba
=a=
"bac a b"