

Q → Given a string, reverse the substring from l to r .

$s = "a b d e a g f"$ $l = 2$ $r = 5$

```

while (l < r) {
    swap(s[l], s[r])
    l++    r--
}

```

TC = $O(N)$ ✓

SC = $O(1)$ → $O(N)$ use String Builder in case of immutable strings.

Q → Given a character array storing a sentence.

Rippling Reverse it word by word.

Takion Corp a) No extra space is allowed, SC = $O(1)$

Paga b) Every word is separated by only one white space ' '.
 c) No leading or ending spaces are present in input.

$A = [\overset{1}{t} \overset{2}{h} \overset{3}{e} \overset{4}{s} \overset{5}{k} \overset{6}{y} \overset{7}{i} \overset{8}{s} \overset{9}{b} \overset{10}{l} \overset{11}{u} \overset{12}{e}]$

Are you as clever as I am
 am I as clever as you Are

1) Reverse complete array. ← TC = $O(N)$

$A = [\overset{0}{t} \overset{1}{h} \overset{2}{e} \overset{3}{s} \overset{4}{k} \overset{5}{y} \overset{6}{i} \overset{7}{s} \overset{8}{b} \overset{9}{l} \overset{10}{u} \overset{11}{e}]$

2) Reverse all words.

reverse(A, l, r-1)

```

l = 0
for r → 0 to N-1
    if (A[r] == ' ')
        reverse(A, l, r-1)
        l = r+1
reverse(A, l, N-1)

```

$TC = O(N^2) \times$
 $\Sigma \text{ length of all words} = O(N)$
 $TC = \underline{O(N)}$
 $SC = \underline{O(1)}$

A = [t h e s k y i s b l u e]
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
 x x x x x x x x x x x x x x x x r = N
 e u l b ↑ s i ↑ y k s ↑ e h t
 t t t l
 b l u e i s s k y t h e

```

l = 0
for r → 0 to N-1
    if (A[r] == ' ')
        reverse(A, l, r-1) ← (r-1) - l + 1 = r - l
        l = r+1
reverse(A, l, N-1)

```

↓ ↓ ↓ ↓ ↓
 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
 t x t x t x t x l x

Q → Given a string of length N.

Return the length of longest palindromic substring.

s = "a b a c a b"
 0 1 2 3 4 5
 0 1 2 3 4
 Ans = 5

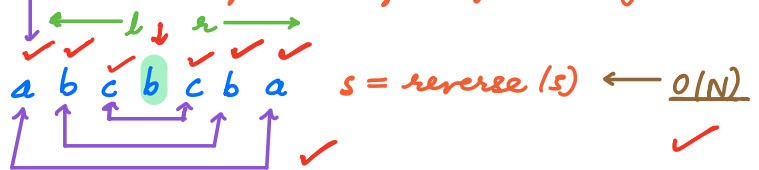
s = reverse(s)
 Eg → "mom", "niten", "dad", etc.

'A' ≠ 'a' Ama ×

s = "happy"
 0 1 2 3 4
 Ans = 2

Bruteforce → \forall substrings, check if it is a palindrome & keep track of longest length.

$$\# \text{ substring} = \frac{N*(N+1)}{2}$$



$$\text{Total TC} = O(N^2 * N) = O(N^3)$$

$$SC = O(1)$$

ans = 0

for $i \rightarrow 0$ to $(N-1)$ // $i \rightarrow$ middle (odd) ✓
 // $i, (i+1) \rightarrow$ middle (even)

$$\text{Total TC} = O(N^3)$$

ans = max(ans, longestPalindromeLength(s, i, i))

if ($i < (N-1)$)

ans = max(ans, longestPalindromeLength(s, i, $i+1$))

return ans

$s = \text{"abacab"}$
 0 1 2 3 4 5
 1 3 1 5 1 1

$s = \text{"abacab"}$
 0 1 2 3 4 5
 0 0 0 0 0

$s = \text{"abbccbbcc"}$
 i i+1
 0 1 2 3 4 5 6 7 8 9

$s = \text{"abcdef"}$