SCALER

String Basics

TABLE OF CONTENTS

1. Strings
2. Questions on Strings

DSA 1 Module ✓

28th June & 1st July → substitute instructor ✓

String

order of characters matter

Array of characters ✓

Group of characters ✗ abc ≠ bac

Sequence of characters ✓

Character

Single symbol that represents →

1) letter / alphabet
2) digit
3) special symbol ('#', '_', etc)

How to store characters?

Every character has a ASCII value mapped.

[0 255]

can be stored in

8 bits = 1Byte ✓

'A' — 'Z' = 65 — 90

'a' — 'z' = 97 — 122

'0' — '9' = 48 — 57

1. char ch = 'a';

    print(ch); → a

2. int ch = 'a';

    print(ch); → 97

3. int ch = 'a' + 1; 97 + 1

    print(ch); → 98

4. char ch = (char) ('a' + 1) (char) (98)

    print (ch) → b

double x = 98 ; ✓

int x = 2.8 ; ✗

Switch Case

< Question > : Given a string consisting of lower-case and upper-case alphabets.

Convert: (1) lowercase → uppercase

(2) uppercase → lowercase

1. **"Hello"** - "hELLO"

2. **"aDgbHJe"** - "AdGBhjE"

'a'	97	'A'	65	97 - 65 = 32
'b'	98	'B'	66	98 - 66 = 32
'c'	99	'C'	67	99 - 67 = 32
⋮		⋮		
'z'	122	'Z'	90	122 - 90 = 32

```
N = A.size()
for i → 0 to (N-1) {
    ch = A[i]          // A.charAt(i)
    if ('a' <= ch && ch <= 'z')  // 97 <= ch <= 122
        A[i] = (char)(ch - 32)
    else
        A[i] = (char)(ch + 32)
}
```

TC = O(N)   SC = O(1)

In some languages updating char in string is not allowed (eg. Java, Python, etc).

⇒ create another string.

```
ans = " "
N = A. size ()
for i → 0 to (N-1) {
    ch = A[i]            // A.charAt(i)
    if ('a' <= ch && ch <= 'z')     // 97 <= ch <= 122
        ans + = (char) (ch - 32)
    else            > appends
        ans + = (char) (ch + 32)
} return ans
```

TC = O(N)   SC = O(1)

→ C++ ✓

→ Java ✗ TC = O(N²)

appending char in string → TC = O(length)

To solve → 1) convert string to char array

2) Solve using char array

3) convert char array back to string

SC = O(N)   TC = O(N)

Substring

1. Contiguous part of a string. ✓
2. A single character is also a substring. ✓
3. Whole string is also a substring. ✓
4. Empty string (" ") is not a substring. ✓
5. String of length N. How many substrings will be there? → $\frac{N*(N+1)}{2}$

a b c

a
a b
a b c
b
b c
c

Check for substring if it's a palindrome or not.

str = reverse (str)

si ei

str - a , b , m , a , d , a , m , t , a , m   si = 2 , ei = 6

0 1 2 3 4 5 6 7 8. 9

Ans = true

```
boolean isPalindrome( String str, int si, int ei ) {

    l = si      r = ei
    while ( l < r ) {
        if ( str[l] ! = str[r] )
            return false
        l++
        r--
    }
    return true

}
```

TC = O(N)   SC = O(1)

< Question > : Given a string s. Find the length of the longest palindrome substring in s.

s → "a n m a d a m m" 1 ≤ N ≤ 10^3

Ans = 5

"f e a c a b a c a b g f"

Ans = 7

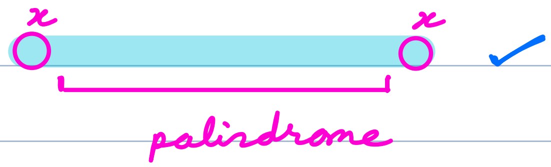
"a d a e b c d f d c b e t g g t e"

Ans = 9

Bruteforce → ∀ substring check if it is palindrome.

$TC = O(N^3)$ $SC = O(1)$

Sol → ∀ char / 2 chars as middle, expand the substring till it is palindrome.



odd length → 1 middle

even length → 2 middle char

[Idea - 2] - "a , d , a , e , b , c , d , f , d , c , b , e , t , g , g , t , e "

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

odd

"a d a e b c d f d c b e t g g t e"

x x

ans = ~~1~~ ~~3~~ ~~5~~ ~~7~~ 9   len = ~~1~~ ~~3~~ ~~5~~ ~~7~~ 9

even

"a d a e b c d f d c b e t g g t e"

ans = ~~2~~ ~~4~~ 6   len = ~~2~~ ~~4~~ 6

final ans = max (9, 6) = 9

"a b c c b c"   Ans = 4

</> Code

```
oddAns = 0
for i → 0 to (N-1) {        // middle
    l = r = i
    len = 1
    while (l >= 0 && r < N && s[l] == s[r]) {
        oddAns = max (oddAns, len)
        l--    r++
        len += 2        // r - l + 1 = len
    }
}
evenAns = 0
for i → 1 to (N-1) {        // (i-1, i)
    l = i-1        r = i
    len = 2
    while (l >= 0 && r < N && s[l] == s[r]) {
        even Ans = max (evenAns, len)
        l--    r++
        len += 2        // r - l + 1 = len
    }
}
return max (oddAns, evenAns)
```

TC = $O(N^2)$ SC = $O(1)$

Immutability of the strings

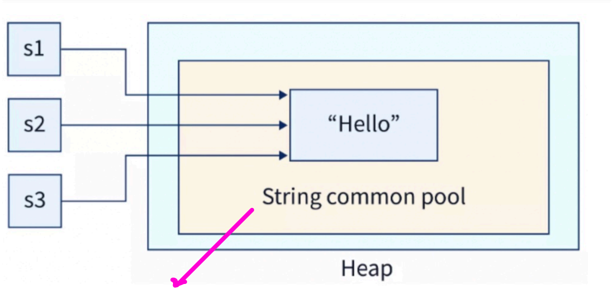
(java, python, C#, golang, etc)

→ value cannot change

String s1 → "Hello"

String s2 → "Hello"

String s3 = s1



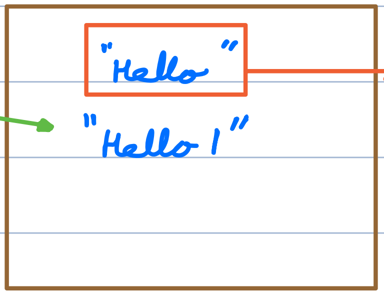
reusable ⇒ save memory

String a = "Hello"

a = a + '1'

// new string is created.

TC = O(length)



→ garbage collector free up the memory.

How to update efficiently ?

1) convert string to array → update → convert back

2) Mutable form in some languages ?

Eg → StringBuilder in Java

③

string →① StringBuilder ②(complete steps) sb.append('1') ← TC = $\underline{O(1)}$