

String Manipulation

Building Blocks of characters
String = array of characters Unicode

Button on Keyboard :

'a' - 'z'

'A' - 'Z'

0's

'@', '!'

'0' - '9'

'.' - '!'

☺ ☺

Represent characters as 0's and 1's

(ASCII)

c/c++ 1 character \rightarrow 1 byte \rightarrow 8 bits \rightarrow numbers.
1 byte \rightarrow can represent 256 chars

[0 - 255]

Char \rightarrow [0, 255]

ASCII: American standard code for information interchange.

Every char \rightarrow Assigned ASCII [0 - 255]

'a' - 'z'
 \downarrow
97
= , .

'A' - 'Z'
 \downarrow
65
@ !

[0, 255]

'O' - 'a':
↓ ↓
49 51

Perform Arithmetic operations

(int) num = a ; // num = 97
char ch = $a + 1$ 98 / b?
↓ ↓
98 → char. ⇒ 'b':
, loword

(char ch); (97 - 122)
, if $ch \geq ('a')$ || $ch \leq ('z')$ (10, 128)
Lower case

1 Byte / 2 Bytes ⇒ (256) is 64,436

Devanagire, kannada, Emojis, tamil
Unicode [n bytes] \rightarrow (10)
(???)

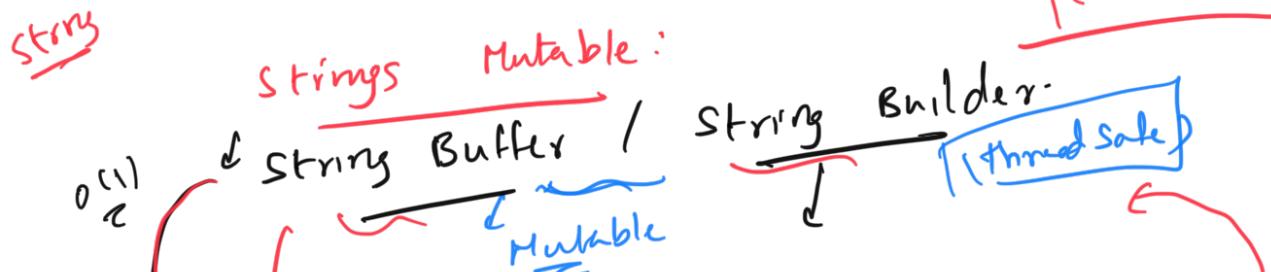
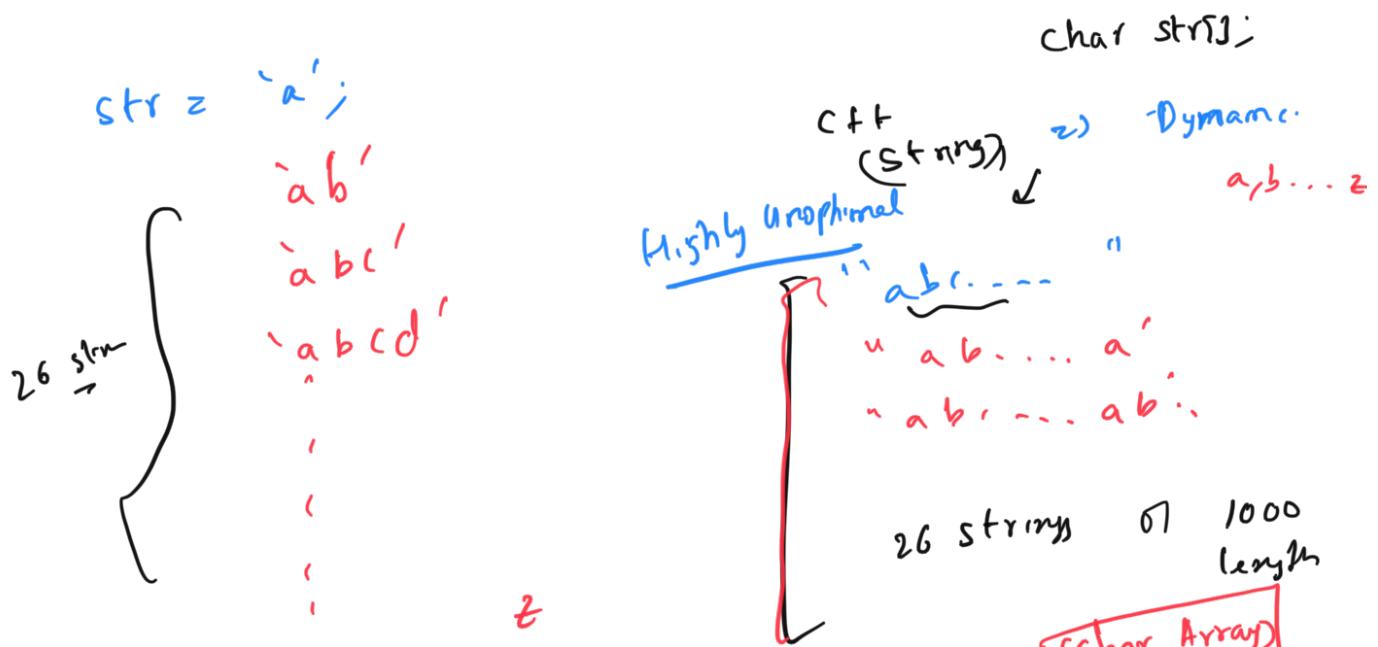
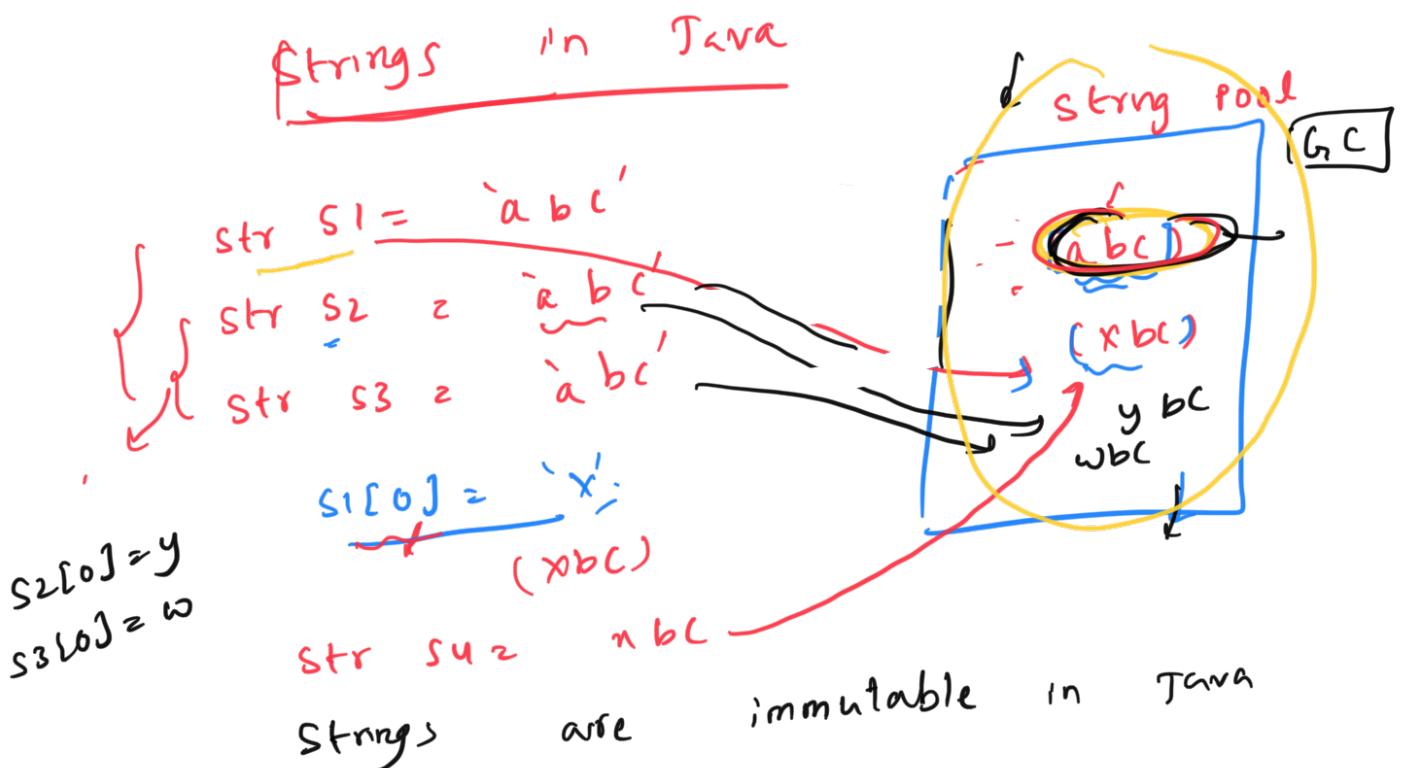
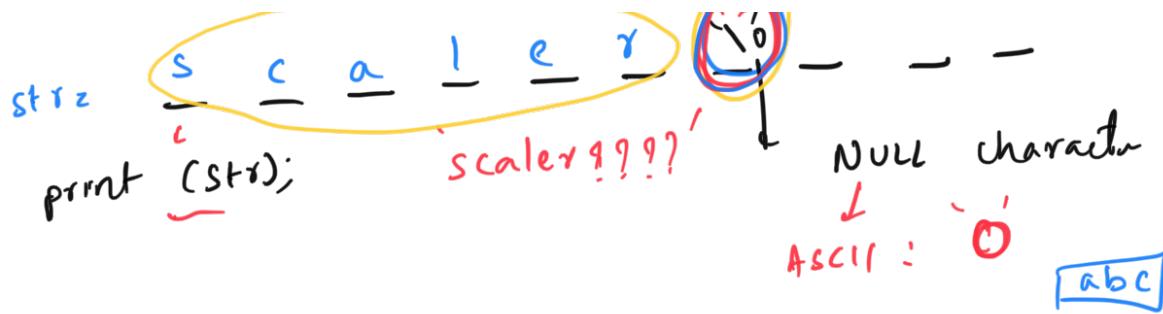
② Unicode trees getting updated.

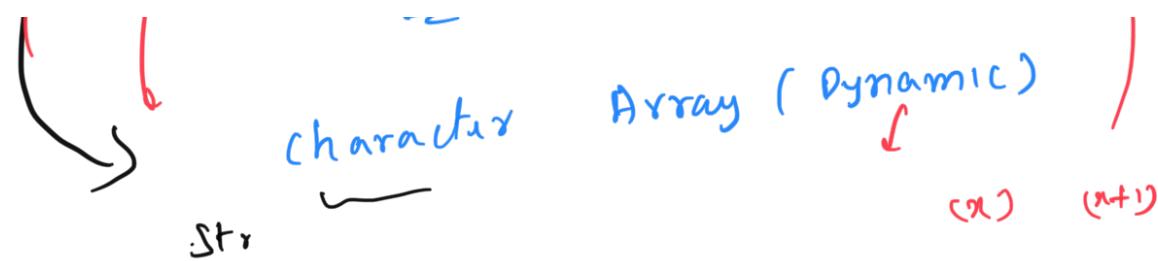
Strings: Array of characters

C++ char str[10]; (5)
'scalar' using
6 character

String Builder
String Build
'10'
character array
5 ×
10 characters
→

char str[10]





Builder pattern \Rightarrow LLD classes:

Question: Reverse a string \Rightarrow (Reverse an array)

str = "SCALER", $\Rightarrow O(1)$ $\frac{n}{2}$ swap

String Builder

```

for(i=0; i< n/2; i++) {
    swap(str[i], str[n-i-1]);
}

```

T.C: $O(n)$
S.C: $O(1)$

substring

contiguous part

of string

a (b c d e f) g h

.split(...)

Subsequence,

Order Matter

Subset:

- Order is not important.
- No Duplicates

String Builder
String Builder

String (ST)

↓ character

string str = ...

Question:

Ex: \Rightarrow Reverse a string word by word

"This is a scalar". "relacs si si,!!"

str(i,j)
(i, (n))

\Rightarrow "Scalar" is
- ["This", "is", "scalar"]
This is "scalar" by Pseudocode

Approach 1:

Array / Vector of strings $\rightarrow O(n)$

$\text{arr} = [\text{"This"}, \text{"is"}, \text{"Scalar"}]$ + $O(n)$

$\text{str} = \text{"Scalar is This"}$

T.C: $O(n)$
S.C: $O(n)$

$\text{Inp} = \text{"Scalar is This"}$

$\text{Out} = \text{"Scalar is This"}$

append "

Inbuilt Functions
split, strip
substring, length

$\text{str} = \text{abc def}$

char
↓
abc def...
1) length
2) substring
append

as Input

$O(n)$

$\text{cout} \ll \text{" "};$
 $\text{print(" ")};$

(spht, strp)
(substring, length)

Approach 2:

$\text{Inp} = \text{"this is scalar"}$

$\rightarrow \text{Rev}(\text{Inp}) = \text{"scalar is this"}$ $\rightarrow O(n)$

$\text{Expected} = \text{"Scalar is this"}$

\rightarrow Reverse the words! $\rightarrow O(n)$

$\text{Outp} = \text{"Scalar is this"}$

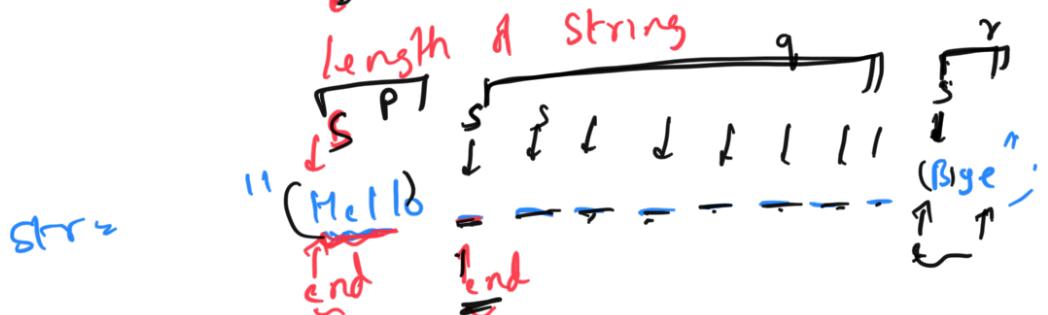
(L, R) string build n

T.C: $O(n) + O(n)$

S.C: $O(1)$

$O(n) \Rightarrow$

str =



Operation: $\underbrace{s + s}_{\text{ }} + 8 + 3 + 3$

split :: $O(n)$

$O(n)$

T.C: $2p + 2q + 2r \approx 2(p+q+r)$
 $= 2n$

$p+q+r=?$

$\boxed{T.C: O(n)}$

Question: Check if a string is palindrome

madam
wow
hannah
Naman
nitin
Anna

T.C, S.C

Approach 1:

str = 'Naman'
rev-str = 'naman'
 $n/2$

$0, n-1$
 $1, n-2$
 $2, n-3$

T.C: $O(n^2)$
 S.C: $O(nm)$

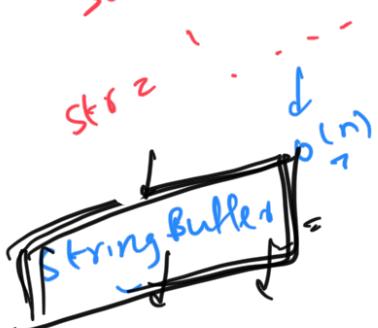
Approach 2:

$\left[\begin{matrix} a & b & c & d & e & c & b & a \\ \downarrow & \downarrow \\ a & b & c & d & e & c & b & a \end{matrix} \right]$

$i, n-i-1$

2 loops

insert() $O(1)$
 swap



T.C:
 $\text{for } (i=0; i < \frac{n}{2}; i++) \{$
 $\quad \text{if } (s[i] \neq s[n-i-1]) \}$
 $\quad \text{return false;}$

\downarrow
 Return true;

S.C:

(3 hrs -)

Question:

Check if it is possible to make a palindrome by changing one character
 string exactly

(Not Deleting / swapping)

Ex: $\left[\begin{matrix} a & b & b & e \\ \downarrow & \downarrow & \downarrow & \downarrow \end{matrix} \right] \Rightarrow abba$

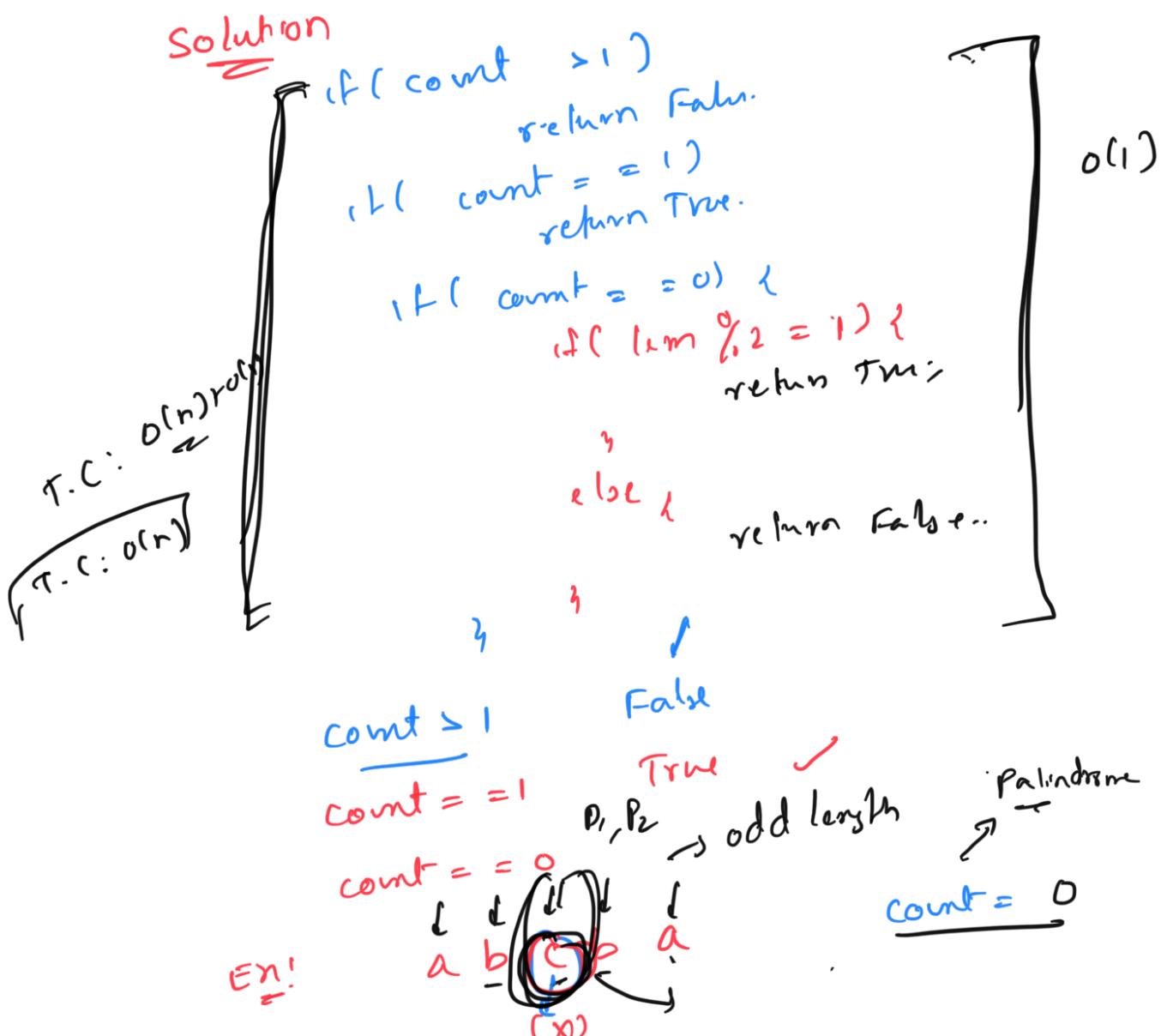
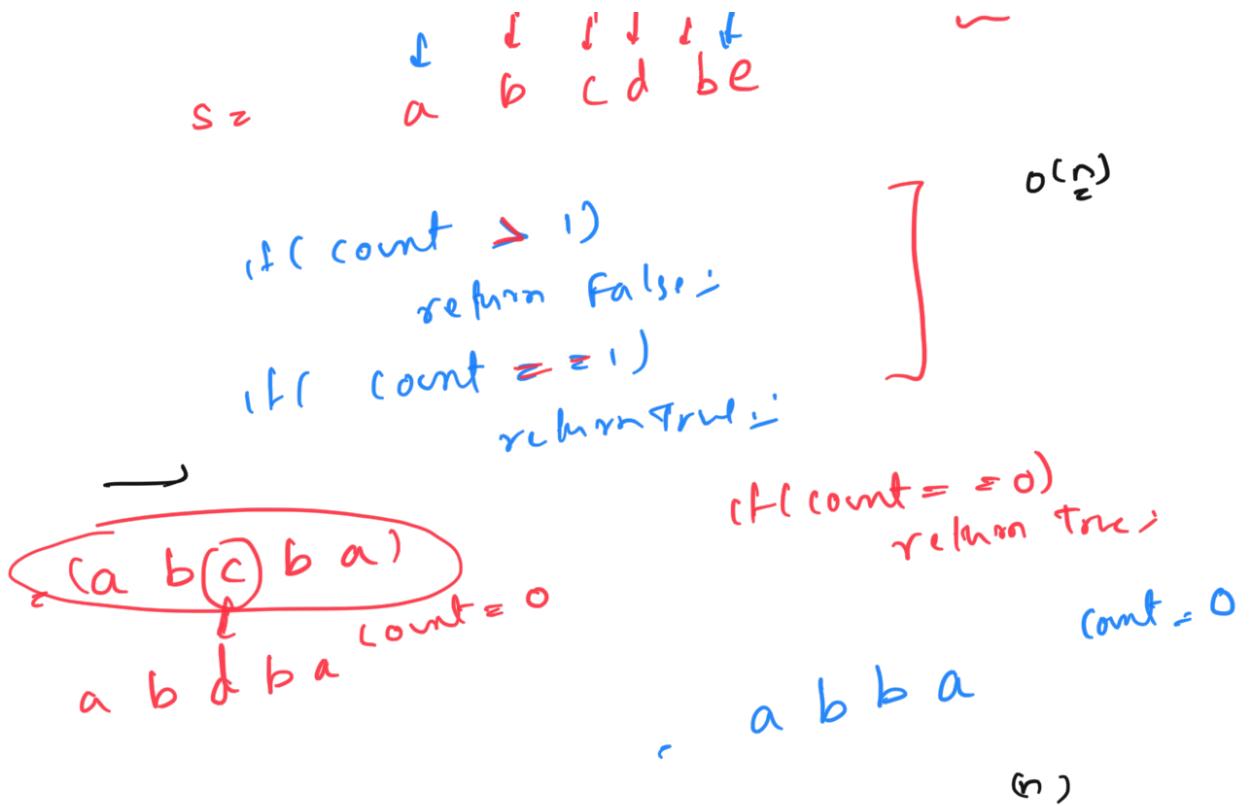
(20)

Ex: $\text{NAMIN} \Rightarrow \text{NAMAN}$

$\boxed{\text{Count = 1}}$

Approach 1 : $\left[\begin{matrix} p_1 & l_1 & t & l_2 & p_2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ N & A & M & I & N \\ p_1 & l_1 & R & l_2 & p_2 \end{matrix} \right]$

Count = 1/2



Ex:  count = 0

odd len Palindrome: True
Even: False

$a b c d b a$
count = 1
 $(a b c c' b a) \Rightarrow$ count = 0
Even length
 \downarrow
 $a b c d b a$

Lexicographical sorted order.
Alphabetic / Dictionary

$A \rightarrow [$ A a
A b
A c

$B \rightarrow [$ B a
B b
B c.

$C \rightarrow [$

$\stackrel{lll}{a a a a a}, \stackrel{lll}{a a a}$

↓
Corry before.

$\stackrel{d}{\text{a a a a b b c}}$

↓
za

Question: Obtain the lexicographically largest string by atmost 1 swap {0, 1} swap

Ex: "string" $\xrightarrow{(1)}$ "t~~s~~ring" { "string" } $\xrightarrow{(0, 1) \text{ swap}}$ "string" { "string" }

a b c d e f g h i j k l m n o p q
 r s t u v w x y z

Ex: "efbs" $\xrightarrow{(5)}$ "sfbe"

Ex: "zabc"

Approach 1:

\rightarrow Find Largest character : $O(n)$
 $\{0, 1\}$

Ex: z a b c
 Largest = z

z c b a

z y a b c
 (Largest = z
 2 largest = y)

z y c b a

Selection Sort.

T-C: $O(n^2)$

Ex: d c b a \Rightarrow 0 swap

Approach 2:

atmost 1 swap

strz $z \ y \ a \ b \ c \ x$
 sort the string (Descending)
 Orig: $z \ y \ a \ b \ c \ x$
 Sort: $z \ y \ x \ c \ b \ a$

Find position of x and swap (Hashmap) $O(n)$
 $O(n)$
 $S.\text{index}(x)$

→ Find First mismatch
 → swap the char $'a'$ at position of $'x'$:
 → Place $'a'$ at

$O(n)$ S.C

Orig: $z \ y \ a \ b \ x \ c \ a \ x$
 sorted: $z \ y \ x \ b \ a \ c \ a \ x$

$S1 = z y x b a c a x$
 $S2 = z y x b x c a a$

Last occurrence
 generate from back & find first

strz $a \ a \ a \ a \ a \ a$
 sort: $a \ a \ a \ a \ a \ a$

all chars are same

strz $[z \ y \ c \ b \ a] \ y$
 sortz $z \ y \ c \ b \ a$

chars in descending order

group comparison

Sorting

$S1-epl$
 1) Sort the string $\Rightarrow O(n)$

Bucket Sort / Count Sort

mismatch

$O(n)$

↑

- 2) First \dots
 3) Finding last occurrence of character $O(n)$

T.C: $O(n) + O(n) + O(n) = O(n)$

S.C: $\sim O(n)$

Arr = [

] $(1-10^9)$

T.C: $O(n \cdot S)$

S.C: $\sim O(n)$

[$1-500$]

count / bucket sort

] 500

(Hint)

String \rightarrow stringbuilder arr2 [

z y x x b x c a x
 sort, z y x x b x c a x

$\rightarrow (1)$

s1 = z y x b x c a x
 s2 = z y x b x c a x

$\rightarrow (2)$

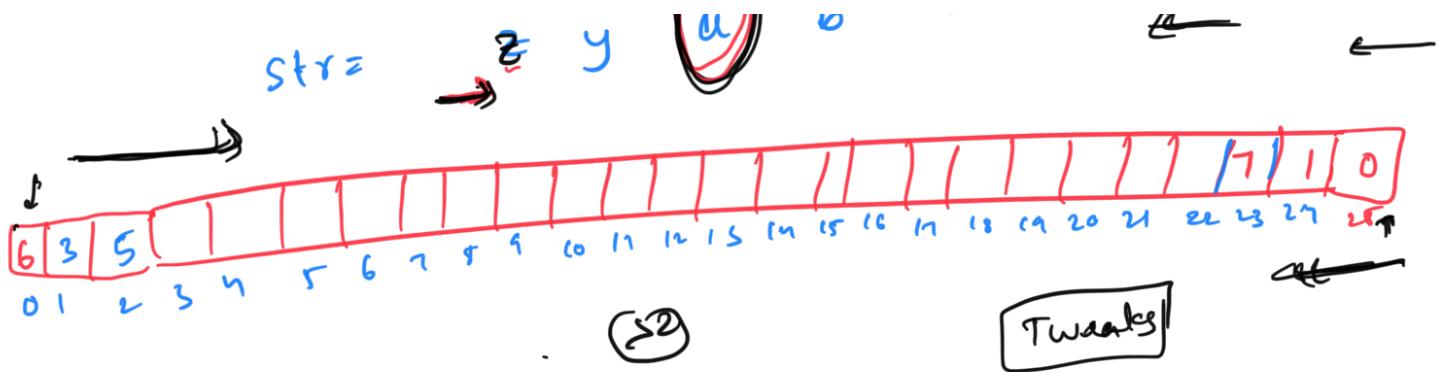
stringbuilder str = new str("abc");

S.C: $O(n)$

0 1 2 3 4 5 6 7 8
 x a c g f d u x
 $\rightarrow O(n)$

$x-a$

$N=8$



$s =$ "this is Scaler"
 $arr =$ "Scaler is this"; $O(n)$
 $rws(s) =$ "relacS.
 ↑
 (s_i)
 ↑
 s
 ↑
 (s'_i)
 ↑
 s
 ↑
 shT
 ↑
 e
 ...e"

" Scaler vs. This" $\Rightarrow O(n)$