

String Manipulation

Building blocks of string: character & array of characters

Character:

Alphabet a - z

A - Z

+/-/!

$\bar{r}' = \bar{q}'$

1

10

o' / 13

Task: Represent char using 0's & 1's

character:

$$1 \text{ byte} = \frac{8 \text{ bits}}{(2^8)} = \underline{\underline{2^5 \text{ B}}}$$

1 byte: C/C++

2 bytes : Java

$\{0-255\} \rightarrow$ represent character | fix

ASCII:

American Standard Code for
Information Interchange

$$\begin{array}{r} \cancel{\text{a}} - \cancel{\text{z}} : \\ \cancel{97} \quad \cancel{122} \\ \hline 65 \end{array} \quad \begin{array}{r} \cancel{\text{A}} - \cancel{\text{Z}} : \\ \cancel{90} \quad \cancel{90} \\ \hline 0 \end{array} \quad \begin{array}{r} \cancel{0} - \cancel{9} \\ \cancel{49} \quad \cancel{57} \\ \hline 1 \end{array} \quad [0, 255]$$

Arithmetic Operations

int num = 'a' + 97
char ch = 'a' + 1

```
// num = 97  
char ch = 'b';  
ch = 'b'  
// ch = 'b';
```

$q_7 + 1 = 48$

→ ch, check if it is lower case

if (ch ≥ 'a' && ch ≤ 'z')
 return true;

(int) ch = ASCII
 1 / 2 bytes $2^{32} \approx 2 \times 10^9$
Unicode [4 bytes] UNICODE!

→ Unicode → not updated.

String
string::size()

string: Array of characters
 (Character Array)

- a) 5
- b) 6
- c) 7
- d) 8

char str[10];

~~~~~  
 'scaler'

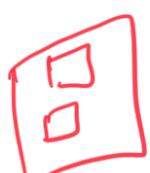
char str[10];

str = s c a l e r ' 0 - - -  
 print(str);

special character: Null character  
 '0'  
 ASCII: 0

scaler ???

||  
 $6 + 1 = 7$   
 ↓  
 Null character

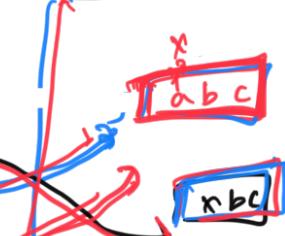


Strings in Java

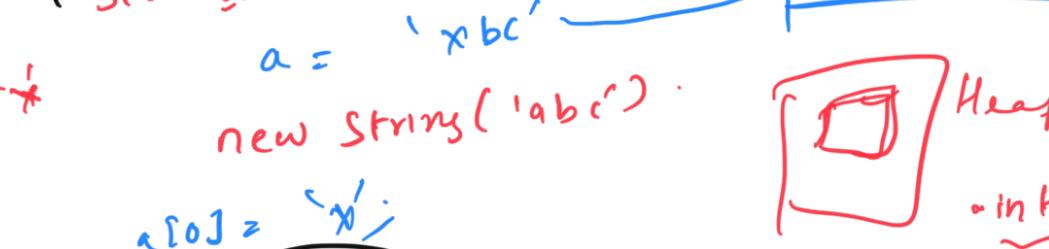
str a =  
 ... i =

'abc'  
 'abc'

String Pool



{  
 Str 1 = 'abc';  
 Str 2 = 'xbc';  
 a = 'abc'  
 new String('abc')  
 a[0] = 'x';  
 a[0] = '\*'  
 a = 'xbc'  
 (Python) ← strings are immutable



a  
 ab  
 abc  
 abcd  
 abcde  
 ...
 abcdef...

z

} 26 strings will be added to root

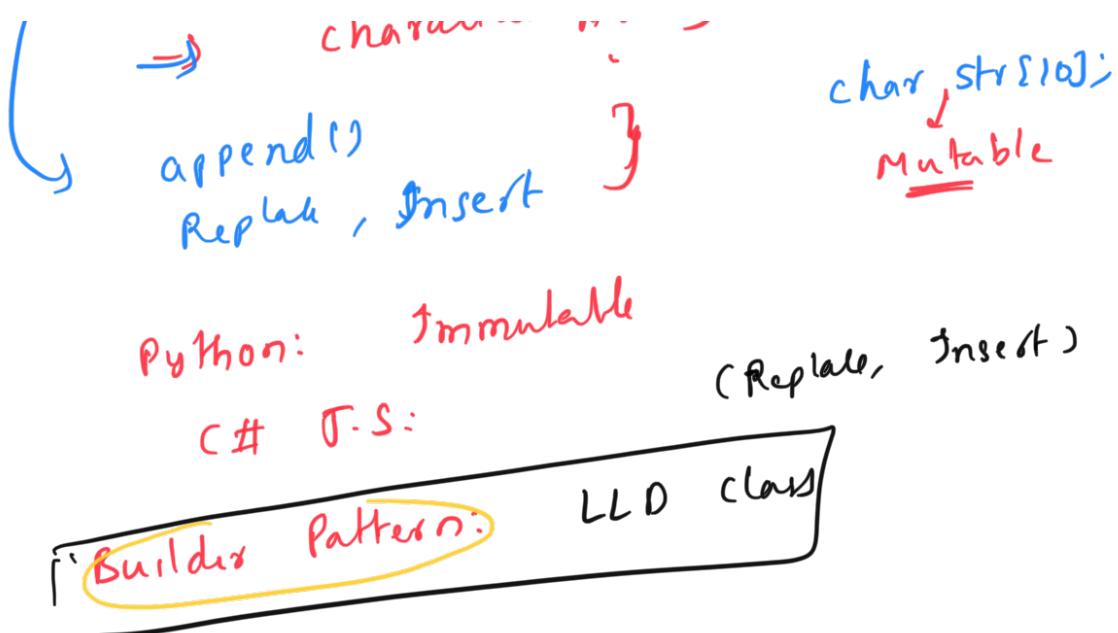
St<sub>8</sub> = " . - - - "

" " " " " " " "

len = 1000

O.S

S  
 Highly Unoptimized  
 with thread safe  
 String  
 Builder / StringBuffer  
 synchronized  
 thread safe  
 Java  
 little slow  
 → strings are mutable  
 String Buffer  
 (26 x 10<sup>00</sup>)



Question: Reverse a String

str = SCALES → RELACS

ans 2

str = "RELAS" (n characters)  
Reverse  
S.C: O(n)

Approach 1

str2 = RELACS      n=6  
swap  $\frac{n}{2}$  characters

str2 = RELACS      0 1 2 3 4 5  
swap      0 n-1  
            1 n-2  
            2 n-3

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

Approach 2: (Stringbuilder)

for(i=0; i <  $\frac{n}{2}$ ; i++)  
swap(str[i], str[n-i-1]);  
S.C: O(1)

**String Builder**

KIDDIN

T.C: O(N)

$O\left(\frac{N}{2}\right)$

string s = "blabla"

ThreadSafe  
O.S

StringBuilder str = string(s) =  
Input string

(Subarray)  
Substring

↓  
Contiguous part  
of a string

Subsequence

↓  
'abcde'

'db'? X

'bd' ✓

Order Matters

(Maths)  
Subsets  
→ Gray order  
→ Unique  
elements.

Reverse

a substring

str = a b c d e f g h  
↑ L R

(Reversing Subarr)

Question: Reverse a string word by word  
→ Input = "This is Scaler" (Split)  
Output = "Scaler is This"

prev(gInput) = "re l a c S si s h T" (S, e)

Approach: str = "t h i s i s (Scaler)"

split the sentence by spaces into words

Array (Vector) of strings

X arr = ["this", "is", "scaler"] ↗

ans = "Scaler\_is\_This"  
..... ↗

$O(n) + O(n)$

↓  
 $O(2n)$

↓  
 $O(n)$

T.C. - 0402

C.C:  $O(n)$

## Approach 2:

Input

 Reverse =  
 Input =

Reversed?

Input = L --- this --- is --- Scale  
{ }  $\Rightarrow$   $\text{main} = z$

this is Scales

"relacs" "S1" "sihT"

"Scales" "is This"

"relacs" "S1" "sihT"

$\rightarrow \text{ST-COL}(n)$   
S.C.:  $\alpha$ )

Reverse this  
substory

(L, R)

$T_c$ :

$$O(n) + O(n) = \underline{\underline{O(n)}}$$

S.C.

$\Omega_2$  (swat)

string Buchstaben

en

$$\left(\frac{5}{2}\right)$$

Input :

$O(n)$

Stringbuilder = 

Input  
Code

$$\text{Rev}(\text{Sty}) =$$

"Scalene right triangle"  $\triangle P_1 P_2 P_3$

q<sub>i</sub>

$$\text{Operations} = (6+6) + \underline{\underline{5}} + \underline{\underline{2}} + (u+u)$$

$$= \boxed{2P_1 + q_1 + 2P_2 + q_2 + 2P_3}$$

$$\begin{aligned}
 & (3P_1 + 3q_1 + 3P_2 + 3q_2 + 3P_3) = \\
 & = 2P_1 + 2q_1 + 2P_2 + 2q_2 + 2P_3 \\
 & = 2(P_1 + q_1 + P_2 + q_2 + P_3)
 \end{aligned}$$

$$O(3n) \approx O(n)$$

$$\left\{ \begin{array}{l} p_1 + q_1 + p_2 + q_2 + p_3 = ? \quad (n) \\ \approx O(2n) \Rightarrow O(n) \end{array} \right.$$

Question: Check if a string is Palindrome

=

str = rev(string)

madam  
nitin  
rayan  
Einel  
lol  
raccar  
malayalan

Approach 1:

str1 = "madam"  
str2 = "madam"

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

S.C:  $O(n)$

Approach 2:

ma dam

0, n-1  
1, n-2  
;  
i, n-i-1

```
for (i=0; i < n/2; i++) {
    if str[i] != str[n-i-1]
        return false
}
```

return True;

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

Question: Make a palindrome  
 check if its possible to make this  
 string a palindrome by changing  
 exactly one character

Ex:  $\begin{array}{cccccc} a & b & b & e \\ \downarrow & & \downarrow & \\ e & & a & \end{array} \Rightarrow abba$   
 1 char

Ex:  $\begin{array}{cccccc} a & b & c & d & b & a \\ & & \downarrow & & \\ & & c & & & \end{array} \Rightarrow YES$

$\boxed{abccba}$

$\begin{array}{cccccc} a & b & c & d & b & c \\ & & & \downarrow & & \\ a & b & c & d & b & a \end{array}$   
 Nb

$\begin{array}{cccccc} a & b & c & d & b & a \\ \swarrow & \downarrow & \uparrow & \downarrow & \uparrow & \searrow \\ a & b & c & d & b & a \end{array}$

$\boxed{\text{Count} = 0 \mid}$

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

$\text{if}(\text{Count} > 1)$   
FALSE

$\text{if}(\text{Count} == 1) \quad \text{d}$   
TRUE

$\text{Count} = 0 \mid$  palindrome

$\therefore$

$\text{str} = a b \text{C} b a$

$\text{if}(\underline{\text{Count}} = 0)$   
TRUE

Odd length  
T R D E

(a b d o n)

str = a b b a  
Even:  
count 0  
return false

// compute the count  
 if (count > 1)  
 False  
 if (count == 1) }  
 True  
 if (count == 0){  
 if (len is odd)  
 True  
 else

a c c a  
a b b a  
a b c a  
count = 1

False:  
 s ↓ s ↓ f ↓ e ↓  
 a b c b a  
 c c  
 count = 0  
 ab n ba

E<sub>2</sub>  
 L L L ↓  
 a B b a  
 Count = 0  
 s ↓  
 " " A B C  
 Count = 1

odd: True  
even: False

L  
 A B A  
 C B C

Lexicographical order  
 L-hr / Dictionary

## Alphade... ,

$$A \rightarrow \begin{bmatrix} a & a \\ a & b \\ a & c \end{bmatrix}$$

$$B \rightarrow \begin{bmatrix} b & a \\ b & b \\ b & c \end{bmatrix}$$

$$C \rightarrow \begin{bmatrix} \end{bmatrix}$$

"aaa" "aaaaa"  
1<sup>st</sup>                  L<sup>nd</sup>

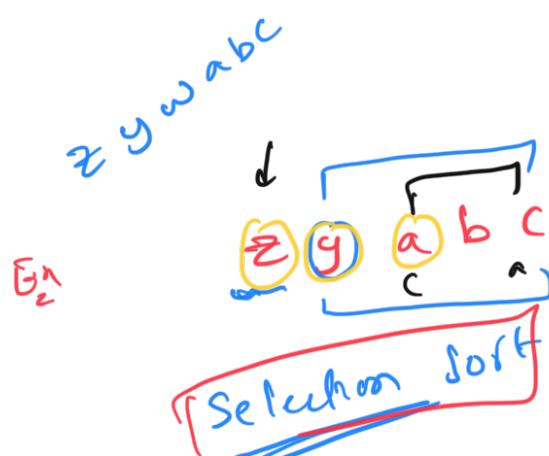
ab  
ac  
az

d  
anaabbcc

Question: Obtain the lexicographically largest string by at most 1 swap.  
(exactly) change)

2) a b c d e f g h i j k l m n o p q r s t  
String:      t s r i n g }  
                      { 0, 13  
                      i t r s n g } X  
                      g t r i n s } X  
                      t s r g n i  
e f b s:      s f b e  
                      . z c b a

$z^a b^c \Rightarrow \sim$

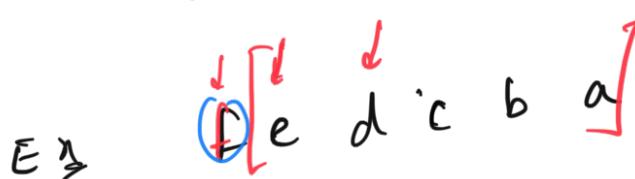


$\sim z$

3rd Largest  $\sim$

$O(n)$

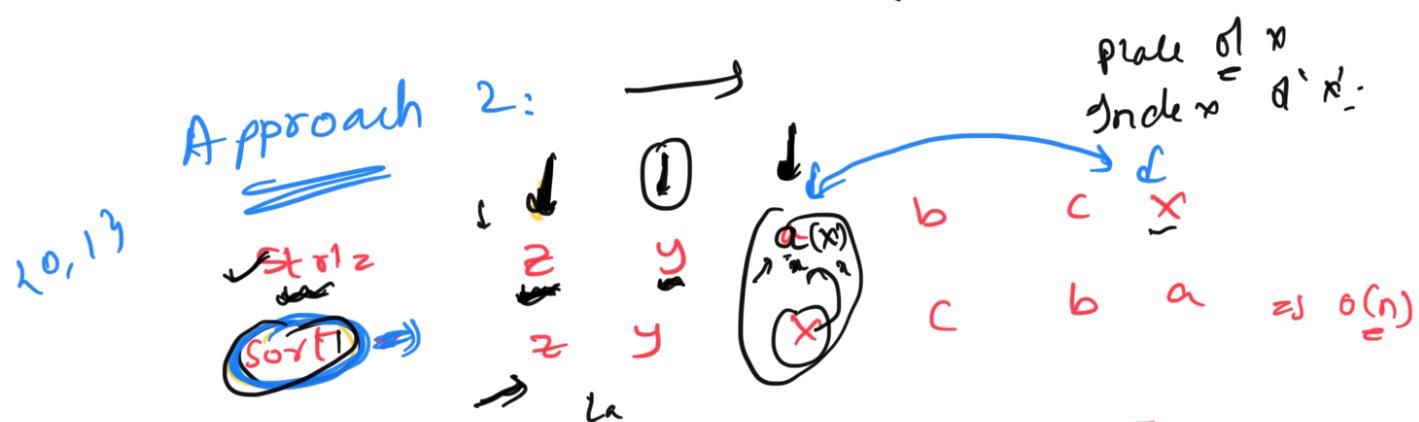
T.C:



$O(n) = O(n)$   
 $O(n) = O(n)$

$n \text{ times}$

$\sim \text{swap}$



place of  $x$   
Index  $i = a, x$

$\approx O(n)$

Count Sort / Bucket Sort  $\rightarrow$

T.C:  $n \log n$   
 $O(n) \Rightarrow O(1) \text{ space}$

Merge Sort  
Quick Sort  
Heap

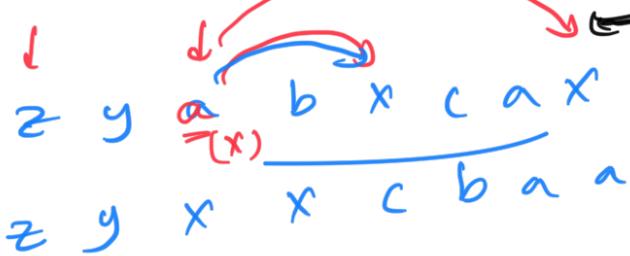
$\approx O(n)$

Algo:

- 1) Sort string in Descending :  $O(n \log n)$
- 2) Find 1st Mismatch :  $O(n)$
- + 1st 'x' in Original

3) Find max v1  
string:  $O(n)$

④ v1:  
**Sort**



Hashmap  
generate from  
back

- 1) z y x b a c a x
- 2) z y x b x c a a ↗

- 1) Sort in v1  $\Rightarrow O(n)$
- 2) First mismatch  $\Rightarrow O(n)$
- 3) generate from back of  
find the last occurrence

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

( $a' - z'$  by  
(26) Poss)

str = a b c d e f g y  
 count = [1 0 3] i 0 o 0 | 1 1 0  
 $\Rightarrow$   $\sum_{i=1}^{26} \text{count}[str[i] - 'a']$   $\boxed{26}$

$O(n)$

$n(n)$

Str = a c c c d x y  
 T.C:  $O(n) + O(n) = O(n)$   
 S.C:  $O(26) \approx O(1)$

will it work for integers?

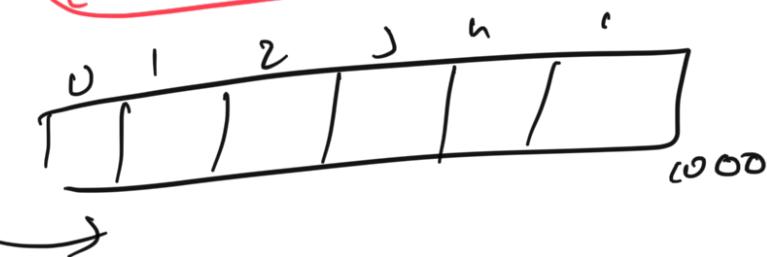


$$1 - 2^{32}$$

$$1 - 10^9$$



$(1 - 1000)$



Memory limit error

StringBuilder =



$$\begin{aligned} 'a' &: 97 \\ 'b' &: 98 \\ 'c' &: 99 \end{aligned}$$

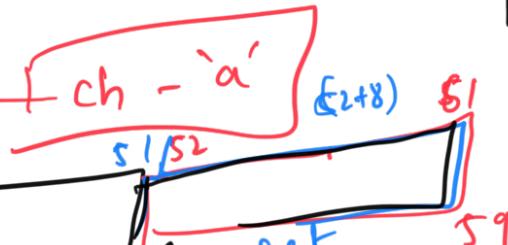
$$('a' - 97)$$

$$('d' - 'a') =$$

$$100 - 97 = 3$$

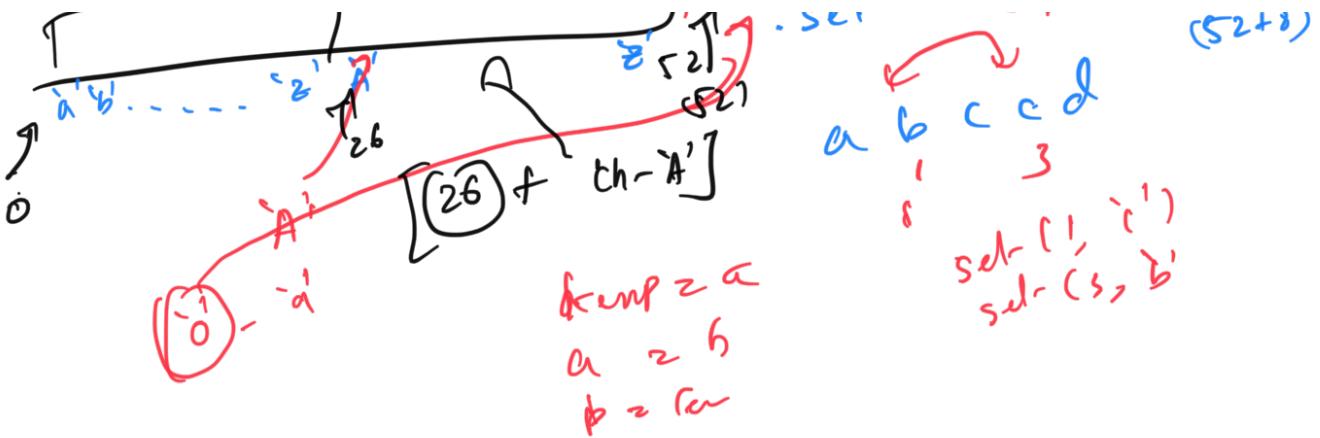
$$\begin{aligned} 'a' - 81 \\ 'A' - 21 \end{aligned}$$

$$ch - 'a'$$



Lower Case:

$$\begin{aligned} ['8' - '0'] &= 8 \\ 824 & \end{aligned}$$



$\text{str} = (\text{a } \text{b } \text{c } \text{d } \text{b } \text{c})$   $\text{count} = 0$

$\text{max} - \text{min} \leq 10^6$   
 $\downarrow$   
Max Array Size

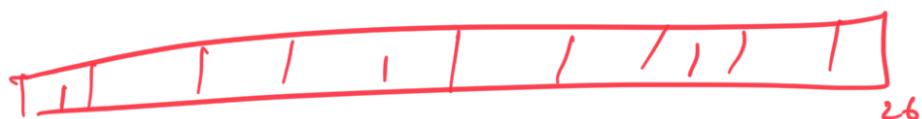
$\text{min} = 1000$   
 $\text{max} = 10^9$

$\text{max} - \text{min} \leq 10^6$   
 $\downarrow$   
Count Sort

Max Size Array:  $[10^6 - 10^7]$

$(\text{a} - \cancel{\text{z}})$   
 $\downarrow$   
26 Possible

"n-a\_d e f h i"



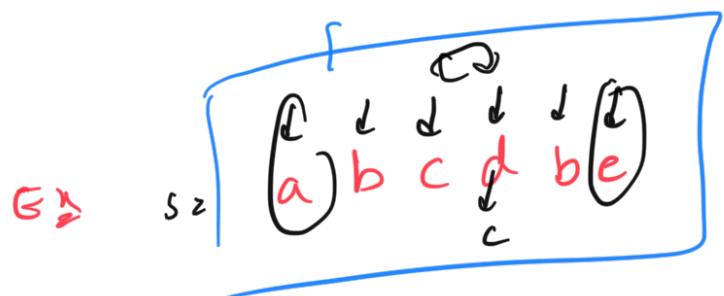
Array of Integers  $[0 - 10^9]$



Arr  $[10^9]$  ? x

Array of Integers  $[1 - 1000]$

Time -



Time -

$$\begin{aligned} \text{count} &= 0 \\ &= 1 + 1 \\ &= 2 \quad \text{false} \end{aligned}$$

Count 1