

Strings

- Array of characters ✓
- Group " " ✗
- Set " " ✗
- Sequence " " ✓

Slack → Deepak Kasera
 deepak.kasera-1@
Scalr.com

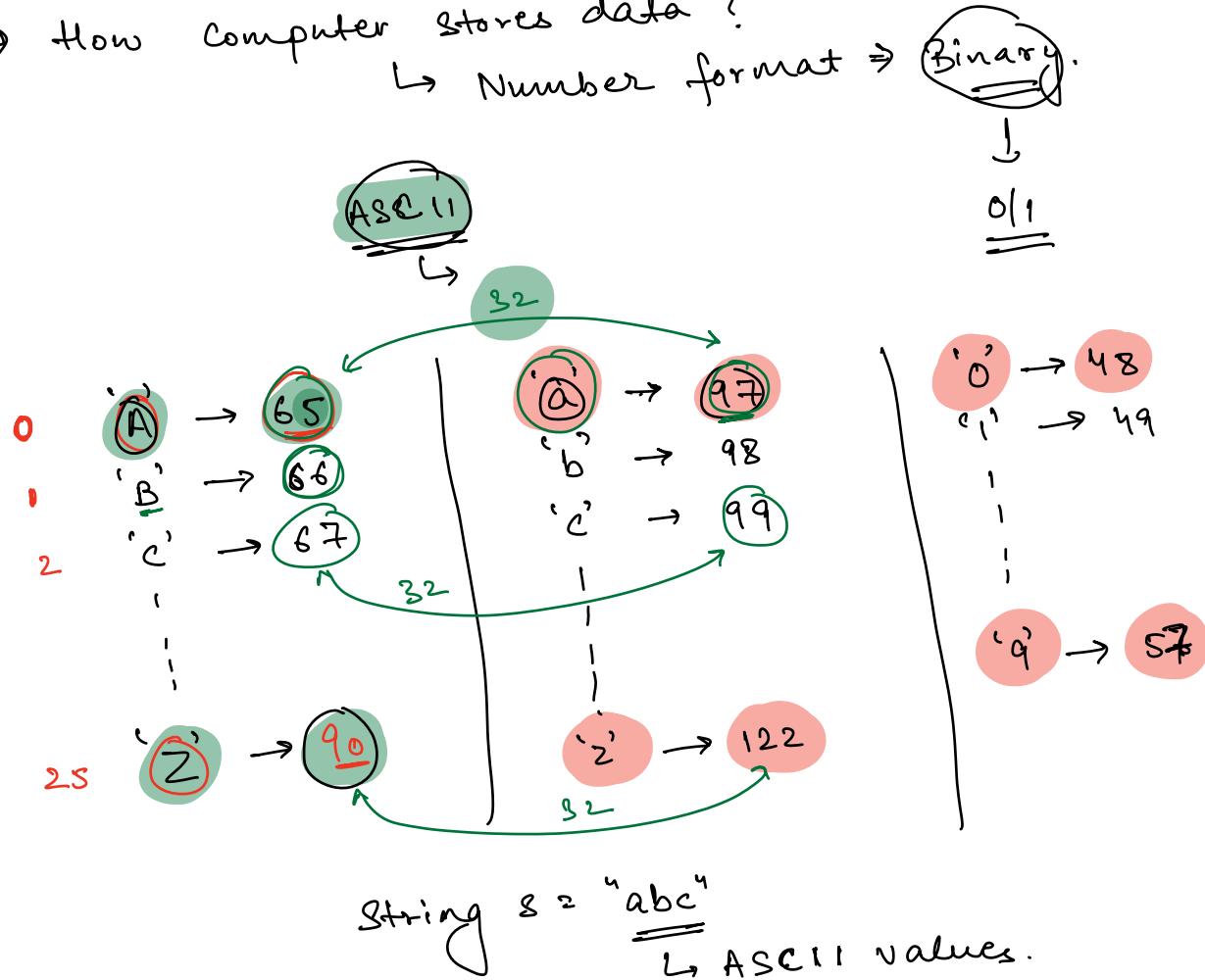
$$\rightarrow S_1 = "abcd" \quad \boxed{f} \rightarrow \times$$

$$\rightarrow S_2 = "bcda" \quad \boxed{f}$$

→ String is a **ordered sequence** of characters.

⇒ How Computer stores data?

↳ Number format ⇒ Binary.



Q Given a String s , toggle the case of every character.

Small case \rightarrow Upper case
Upper case \rightarrow Small case.

$s = "aA bB cC" \rightarrow "AaBbCc".$

- $s[i]$ → check if this character is small / upper case.
- 1) $s[i] \geq 65 \text{ & } s[i] \leq 90 \Rightarrow$ Upper case.
2) $s[i] \geq 97 \text{ & } s[i] \leq 122 \Rightarrow$ Lower case.

```
String toggle(string s) {
    for (i=0; i<s.size(); i++) {
        //  $s[i]$  is a small case
        if ( $s[i] \geq 'a'$  &&  $s[i] \leq 'z'$ ) {
            s[i] -= 32; //  $a - 'A' / 'b - 'B' / \dots$ 
        }
        //  $s[i]$  is a upper case.
        else if ( $s[i] \geq 'A'$  &&  $s[i] \leq 'Z'$ ) {
            s[i] += 32;
        }
    }
    return str;
}
```

$'z' \rightarrow \underline{90} \Rightarrow 1011010$
 $'z' \rightarrow \underline{122} \Rightarrow 1111010$

$$32 \Rightarrow \underline{\underline{1000000}}$$

(XOR)

$$s[i] \wedge \underline{\underline{32}} \rightarrow 2^5 (1 \ll 5)$$

$\leftarrow 'z' \rightarrow \underline{90}$
 $\leftarrow 'z' \rightarrow \underline{122} \Rightarrow 1111010$

$$'z' \rightarrow 'z'$$

$$+32$$

$+32 \Rightarrow$ Setting the 6th Bit
 $-32 \Rightarrow$ Unsetting the 6th Bit

XOR	$0 \wedge 1 = 1$
	$0 \wedge 0 = 0$
	$1 \wedge 1 = 0$
	$1 \wedge 0 = 1$

$$\begin{array}{r}
 90 \rightarrow 1011010 \\
 32 \rightarrow 0100000 \\
 \hline
 1111010 \\
 0100000 \\
 \hline
 0
 \end{array}$$

Given a string, and string contains only lower case characters. Sort the string

$$S = "dabacedb"$$

$$\hookrightarrow \underline{aabbddde}$$

\Rightarrow sort function $\rightarrow O(N \log N)$

Number of distinct characters \Rightarrow

26

Lower
case.

$S = "dabacedb" \Rightarrow$ frequency array

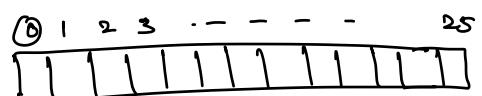
a :	2
b :	2
c :	0
d :	2
e :	1
f :	0
g :	0
:	

\Rightarrow 'aabbdde'
SORTED

1) Count the frequency of every character.

2) Print / Store the characters

Char	ASCII	index
'a'	97	0
'b'	98	1
'c'	99	2
'd'	100	
:		
'z'	122	25



int freq[26] = {0};

$\underbrace{\text{index}}_{\text{index}} \Rightarrow \underline{s[i]} - 'a' \Rightarrow s[i] \Rightarrow \text{Index} + 'a'$

```

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

$\boxed{2}$

$\boxed{\text{||}}$

$\boxed{3}$

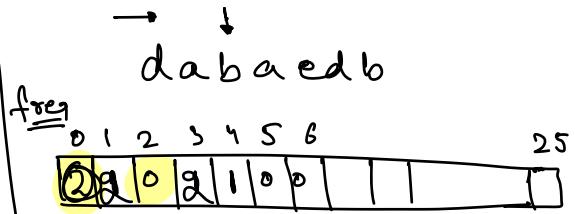
$\boxed{\text{||}}$

$\boxed{3}$

// Sort the string using freq.

```

for (i = 0; i < 26; i++) {
    for (j = 0; j < freq[i]; j++) {
        ans += (index + 'a');
    }
}
    
```



$i=0, \text{index} = 3$
 $i=1, \text{index} = 0$
 $i=2, \text{index} = 1$
 $i=3, \text{index} = 0$
 $i=4, \text{index} = 4$
 $i=5, \text{index} = 3$
 $i=6, \text{index} = 1$

 $i=0, \text{freq}[i] = 2$

TC: $O(N)$

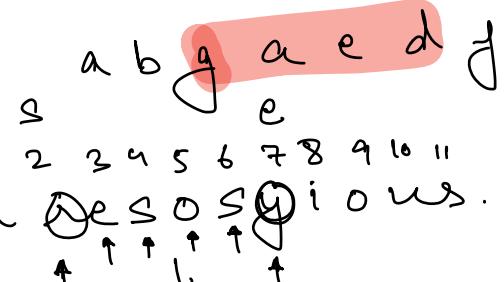
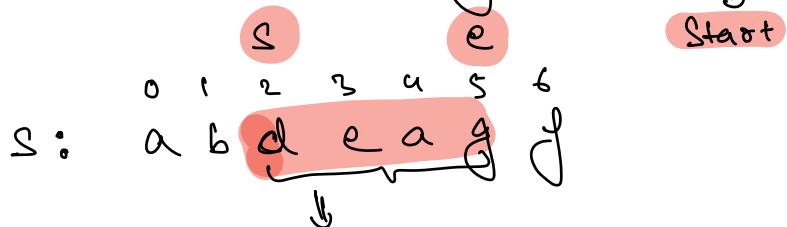
SC: $O(1)$

\Rightarrow Count Sort
 \hookrightarrow Counting freq.

i	freq[i]	
0	<u>freq[0]</u>	1a1
1	<u>freq[1]</u>	1b1
2	1	1c1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1
11	1	1
12	1	1
13	1	1
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	1	1
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1

$1a1 + 1b1 + 1c1 + \dots$
 $\underbrace{\quad\quad\quad}_{\text{Length of String}}$

Q Given a string and start & end indexes.
Reverse the substring from $s[stoe]$ to $s[end]$.



Quiz

reverse (str, s, e) {

while (s < e) {

swap (str[s], str[e])

 s++

 e--

}

}

TC : $O(N)$

SC : $O(1)$

Q Given a String, which stores a sentence

Amazon
MS/UHG

1) → No before/after extra space
2) → Every word is separated by a single space only.

$s = \text{here is a picture}$

Reverse the given string word by word.

o/p

picture_a_is_here

$s = \text{"Are you as clever as I am"}$.



"Am I as clever as you are"

$s = \text{"Mailman brings letters"}$



"Letters brings mailman"

$S = "Malinen brings letters"$ ∈

"Letters brings mailman"

sneltel squi~~r~~b namilam

Diverse
individual
moods

- (I) $\text{reverse}(\text{str}, 0, n-1) \Leftarrow O(N)$
 - (II) Traverse the string, call reverse function on every individual word.

HW: Implement above approach.

Q Given a string, calculate the length of largest palindromic substring.

Amazon |
Directi
flipkart

s : a $\underline{\underline{a \leq a}}$ $\Rightarrow \textcircled{5}$

Palindrome.

s : "abcdef" $\Rightarrow \textcircled{1}$

Quiz

a $\underline{\underline{babab}}$ $\Rightarrow \textcircled{5}$

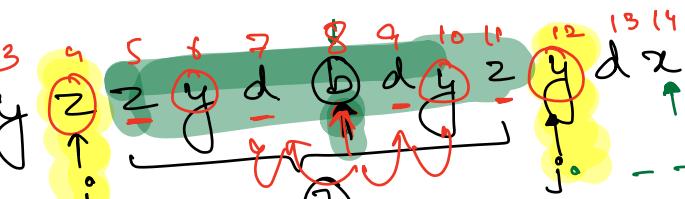
Approach I :-

1. Iterate over all the substrings. $\Rightarrow O(N^2)$

2. Check if a substring is palindrome $\Rightarrow O(N)$

TC : $O(N^3)$

SC : $O(1)$

Quiz :-  $\Rightarrow \textcircled{8}$

$$i = 8, j = 8$$

$$i = 7, j = 9$$

$$i = 6, j = 10$$

$$i = 5, j = 11$$

$$i = 4, j = 12$$

(i, j)

$$\underline{\underline{j - i - 1}}$$

$[a, b]$
$b - a + 1$
(a, b)
$b - a - 1$

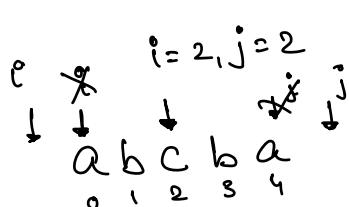
int lengthPalindrome (str, Ci, Cj) $\Rightarrow O(N)$
 $i = C_i, j = C_j$

while ($i > 0 \text{ and } j < n \text{ and } str[i] == str[j]$)
 $i--$
 $j++$

}

return $j - i - 1$

}



\rightarrow for odd length, $C_i = C_j$

\rightarrow for even length, C_i, C_{i+1}

$i = 2, j = 2$
 $i = 1, j = 3$
 $i = 0, j = 4$
 $i = -1, j = 5$

$$5 - (-1) - 1$$

$$5 + 1 - 1 = 5$$

center

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
 $x \text{ } b d y z z y z d \text{ } y z y d x$
 \uparrow
 $i = 4$
 len (str, 4, 5)
 $i = 4 \text{ } ① \text{ } j = 8 \text{ } 9 \text{ } jx$

ans = 0;
 for ($i = 0; i < N; i++$) {

// Check for odd length.

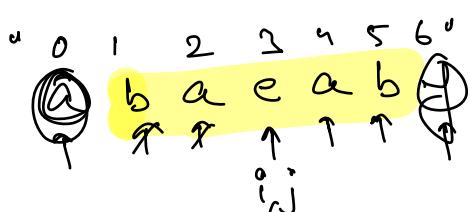
ans = max(ans, lengthPalindrome(str, i, i));

// Check for even length

ans = max(ans, lengthPalindrome(str, $i, i+1$));

}

return ans;



TC: $O(N^2)$

SC: $O(1)$

$$i=0, \text{ fun}(s, 0, 0) \quad \begin{cases} \text{odd} \\ \text{Even} \end{cases}$$

$$\Rightarrow \textcircled{1} \quad -1 \quad 1 \quad \left| \begin{array}{l} \text{fun}(s, 0, 1) \\ 1-0-1=0 \\ = 0 \end{array} \right.$$

$$i=1, \text{ fun}(s, 1, 1) \quad \begin{cases} \text{Even} \\ \text{fun}(s, 1, 2) \\ \Rightarrow 0 \end{cases}$$

$$i=2, \text{ fun}(s, 2, 2) \quad \begin{cases} \text{fun}(s, 2, 3) \\ \textcircled{1} \\ 0 \end{cases}$$

$$i=3, \text{ fun}(s, 3, 3) \quad \begin{cases} \text{fun}(s, 3, 4) \\ \textcircled{2} \end{cases}$$

DP Solution :-

TC $\Rightarrow O(N^2)$

SC $\Rightarrow O(N^2)$

Manacher's Algorithm $\Rightarrow O(N)$ TC