

Today's Quote

Dream big, stay positive, work hard, and enjoy the journey.

Today's content

- Intro
- flip
- sort `ch[i]`
- Is. Palindrome.
- Longest Palindromic Substring

String. → array of characters ✓
 ↳ sequence of characters ✓
 ↳ bunch of characters. ✗

{ a b c }
 { b a c }

★ Not same-order is also important.

characters { ASCII value }

'A' → 65	$\xleftrightarrow{+32}$	'a' → 97
	$\xleftarrow{-32}$	
'B' → 66		'b' → 98
'C' → 67		'c' → 99
'D' → 68		'd' → 100
⋮		⋮
'Y' → 89		'y' → 121
'Z' → 90		'z' → 122

'0' → 48
 '1' → 49
 '2' → 50

'9' → 57

'\0' → It is not a single character.

ch = 'a'

ch = ch + 8

print(ch) ASCII → 65.

'A'

String. → array of characters.

string s = "abcd"

s →

a	b	c	d
---	---	---	---

 0 1 2 3

→ print(s[2])

→ c will get printed.

Q Given a char[], toggle every character.

↳ Capital \rightleftharpoons Small

Note \rightarrow Input contains only small & capital characters.

Eg \rightarrow AnaConDa

o/p \rightarrow aNAcONdA

pseudo-code

$$32 = 2^5.$$

toggleCharacters(s[], N) {

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

if (s[i] >= 65 && s[i] <= 90) {

// s[i] is capital

s[i] += 32

} else {

// s[i] is small

s[i] -= 32

}

}

}

$$s[i] = s[i] \wedge 32$$

$$s[i] = s[i] \wedge (1 \ll 5)$$

T.C $\rightarrow O(N)$
S.C $\rightarrow O(1)$

{unset}

	7	6	5	4	3	2	1	0
A: 65 \rightarrow	0	1	0	0	0	0	0	1
B: 66 \rightarrow	0	1	0	0	0	0	1	0
C: 67 \rightarrow	0	1	0	0	0	0	1	1
⋮								
z: 90 \rightarrow	0	1	0	1	1	0	1	0

{set}

	7	6	5	4	3	2	1	0
a: 97 \rightarrow	0	1	1	0	0	0	0	1
b: 98 \rightarrow	0	1	1	0	0	0	1	0
c: 99 \rightarrow	0	1	1	0	0	0	1	1
⋮								
z: 122 \rightarrow	0	1	1	1	1	0	1	0

Q) Given a char array, which contains only **lower-case alphabets**. Sort given `ch[]` in alphabetical order.

constraints

$$1 \leq N \leq 10^5$$

$$'a' \leq \text{ch}[i] \leq 'z'$$

Ex: `str = d a b a c d b`

↓ After sorting

`str = a a b b c d d`

ideas:

① sort `ch[]` using bubble sort.

$$T.C \rightarrow O(N^2)$$

$$S.C \rightarrow O(1)$$

② Use inbuilt sort function

+
custom comparator
{ if needed }

$$T.C \rightarrow O(N \log N)$$

$$S.C \rightarrow O(1)$$

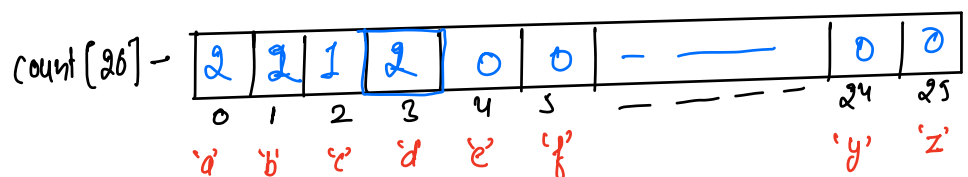
idea-3:

`str = d a b a c d b`
 0 1 2 3 4 5 6

$$\begin{bmatrix} 'a' \rightarrow 2 \\ 'b' \rightarrow 2 \\ 'c' \rightarrow 1 \\ 'd' \rightarrow 2 \end{bmatrix}$$

`str = a a b b c d d`

<u>char</u>	<u>idx.</u>
'a' - 'a' → 0	0
'b' - 'a' → 1	1
'c' - 'a' → 2	2
'd' - 'a' → 3	3
⋮	⋮
'z' - 'a' → 25	25



count[0] \rightarrow frequency of 'a'

count[1] \rightarrow frequency of 'b'

count[2] \rightarrow frequency of 'c'

count[25] \rightarrow frequency of 'z'

step 1 \rightarrow iterate on s[] and fill count[26].

step 2 \rightarrow iterate on count[] & update given char[].

pseudo-code:

sortString (s[], N) {

count[26];

for (i = 0 ; i < N ; i++) {
 idx = s[i] - 'a'
 count[idx] += 1
}

k = 0 // index of s[] array.

for (i = 0 ; i <= 25 ; i++) {
 ch = i + 'a';

for (j = 1 ; j <= count[i] ; j++) {
 s[k] = ch;
 k++
}

}

}

N

T.C $\rightarrow O(N)$
S.C $\rightarrow O(1)$

N

i	j	iters
0	[1, 1]	c[0] +
1	[1, 2]	c[1] +
2	[1, 3]	c[2] +
...
25	[1, 26]	c[25] +
		<u>= N</u>

{ [i, j] = j - i + 1 }

Substring concept is same as sub-array.

- ↳ 1) contiguous part of a string
- 2) full string can be sub-string.
- 3) A single character can be a substring.

Q: check if given substring is palindrome or not.

Ex:

madam	nayan	level
mam	civic	malayalam
dad	radar	

$s=3, e=7$

char ch[11] : { a n a m a d a m e p e }

start index of substring
end index of substring

boolean isPalindrome (str[] , s , e) {

while (s < e) {

if (str[s] != str[e]) {

return false

s++, e--;

}

return true;

}

T.C $\rightarrow O(N)$

S.C $\rightarrow O(1)$

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

Ex: a b a c a b
 0 1 2 3 4 5
 [ans = 5]

Ex: a b c d e
 0 1 2 3 4
 [ans = 1]

idea-1: For all the substrings, check whether they are palindromic or not. Get the max-length.

$$\frac{n(n+1)}{2} * n \Rightarrow$$

$T.C \rightarrow O(N^3)$ $S.C \rightarrow O(1)$
--

constraint: $1 \leq N \leq 3 \times 10^3$.

```

int longestPalindrome ( str (T, N) ) {
    //ans = 0
    for ( i = 0 ; i < N ; i++ ) { //i → start idx of substring
        for ( j = i ; j < N ; j++ ) { //j → end idx of substring
            //substring [i, j]
            if ( isPalindrome ( str, i, j ) ) {
                ans = max ( ans, j - i + 1 );
            }
        }
    }
}
  
```

eg: { x 0 b 1 d 2 y 3 z 4 z 5 y 6 d 7 b 8 d 9 y 10 z 11 y 12 d 13 x 14 }

↑
Centre.

$$\text{length} = (P_1, P_2) \\ = P_2 - P_1 - 1$$

idea: Take any index as the centre & then try to expand to find the length of longest palindromic substring.

$$T.C \rightarrow O(N^2)$$

+

Take any two consecutive indices as centre & then try to expand to find the length of longest palindromic substring.

$$T.C \rightarrow O(N^2)$$

→ max odd length palindrome.

↑ max-length of palindrome.

→ max even length palindrome.

$$\left[\begin{array}{l} T.C \rightarrow O(N^2) \\ S.C \rightarrow O(1) \end{array} \right]$$


```

int expand ( char s[], p1, p2 ) {
    while ( p1 >= 0 && p2 < N && s[p1] == s[p2] ) {
        p1--;
        p2++;
    }
    return p2 - p1 - 1;
}

```

```

int longest Pal ( char[] s, n ) {
    ans = 0/1;

    for ( i = 0; i < N; i++ ) { // max odd length palindromes
        p1 = i, p2 = i;
        ans = max ( ans, expand ( s, p1, p2 ) );
    }

    for ( i = 0; i < N-1; i++ ) { // max even length palindromes
        p1 = i, p2 = i+1;
        ans = max ( ans, expand ( s, p1, p2 ) );
    }

    return ans;
}

```

arr \rightarrow [3, 30, 34, 5, 9]

largest possible no \rightarrow 9534330

$\begin{matrix} 9 & 5 \\ \uparrow & \uparrow \\ \cancel{3} & \cancel{9} \end{matrix}$

9, 5, 34, 3, 30
 $\begin{matrix} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ \text{largest} & 2^{\text{nd}} & 3^{\text{rd}} & 4^{\text{th}} & 5^{\text{th}} \end{matrix}$

Bubble sort

arr \rightarrow [3, ³⁴~~30~~, ⁵~~34~~, ⁹~~30~~, ³⁰~~9~~]

$\begin{matrix} 3034 & 305 & 309 \\ 3430 & \underline{\underline{530}} & \underline{\underline{930}} \end{matrix}$

After 1st iter \rightarrow [3, 34, 5, 9, 30]

Idea 2. Use comparator (a, b)

ab, ba.

{create these two integers}

① Consider all subarrays of $\text{len} = K$, iterate & find max.

$$\text{T.C} \rightarrow (N - K + 1) * K \quad \text{worst case} \rightarrow K = N/2$$

$$\text{T.C} \rightarrow \underline{N^2}.$$

Optimisation \rightarrow Advanced { stack n queue }.