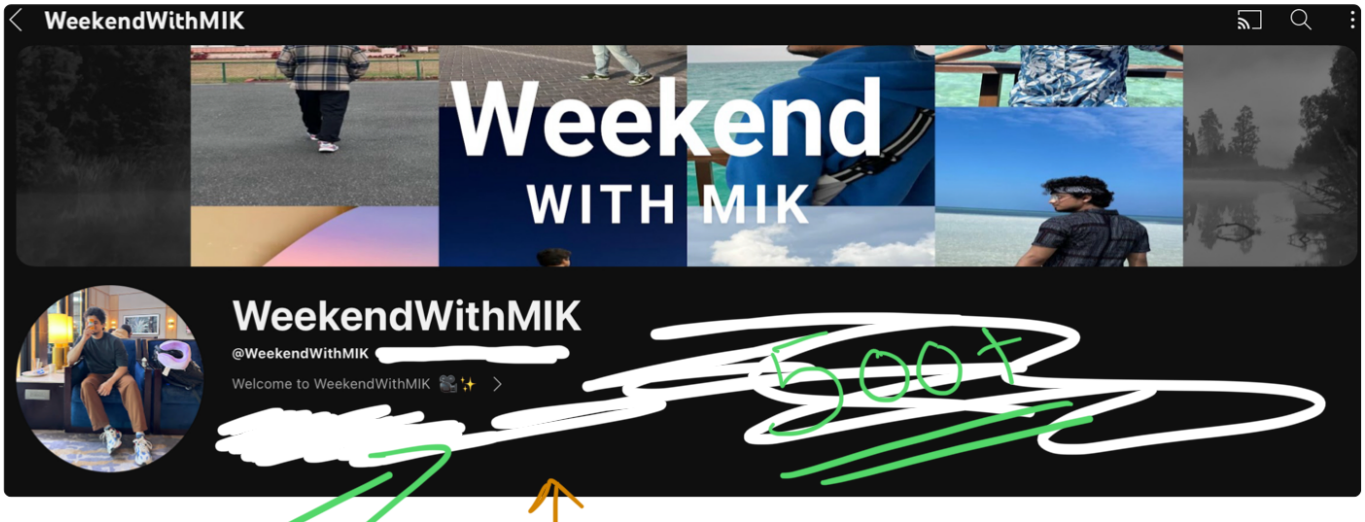
@codestorywithmiK
(Instagram, Facebook)
cswithMIK → Twitter
codestorywithMIK → whatsapp

STRINGS...

Leetcode
- 2981 → Easy
  2982 →
Medium

Video-53



Try this channel to see Life Behind The Scenes ...

Motivation Of the Day :-

You did not come this far to quit.

You don't get what you wish for ;
You get what you work for

MIK...

2982.

2981. Find Longest Special Substring That Occurs Thrice I & II

Medium | Topics | Companies | Hint

You are given a string `s` that consists of lowercase English letters.

A string is called **special** if it is made up of only a single character. For example, the string `"abc"` is not special, whereas the strings `"ddd"`, `"zz"`, and `"f"` are special.

Return *the length of the* ***longest special substring*** *of* `s` *which occurs* ***at least thrice*** *or* `-1` *if no special substring occurs at least thrice.*

thrice or -1 if no special substring occurs at least thrice.

A **substring** is a contiguous **non-empty** sequence of characters within a string.

Example :- S = "aaaa"

"aa"

Output :- 2

S = "abcdef"

'a' 'b'

Output :- -1

S = "abcaba"

Output :- 1

How will approach this problem in an Interview ???

BRUTE Force लगेगा क्या ??? ...

$S = \text{"}\overset{0}{a}\overset{1}{a}\overset{2}{a}\overset{3}{a}\text{"}$

"aaaba"

1. ~~Substrings~~
2. Special Substrings.
3. longest.

{ "a" "aa" "aaa" "aaaa" | "a" "aa" "aaa" | "a" "aa" | "a" } special.

map.

"a"	→ 4 ✓
"aa"	→ 3 ✓
"aaa"	→ 2 ✗
"aaaa"	→ 1 ✗

→ Special substr.

2

Can we optimize further :-

"$\overset{0}{a}\overset{1}{a}\overset{2}{a}\overset{3}{a}$"

"a a a a"
 i j

"aa"

copying the string
O(length of str).

map
<String, int>

~~"a"~~	1+1
~~"aa"~~	1+1
~~"aaa"~~	1
~~"aaaa"~~	1
~~"zzzz"~~	2

map
<pair(char,int), int>

"a"	{'a', 1}	1+1+1+1
"aa"	{'a', 2}	1+1+1
"aaa"	{'a', 3}	1+1
"aaaa"	{'a', 4}	1

T.C = O(n²)

"a a a a"

currChar = 'a' , length = 1

Can we further Optimize

? ? ?

$3 <= s <= 5 \times 10^5$

O(n)

"a a a a"

map.

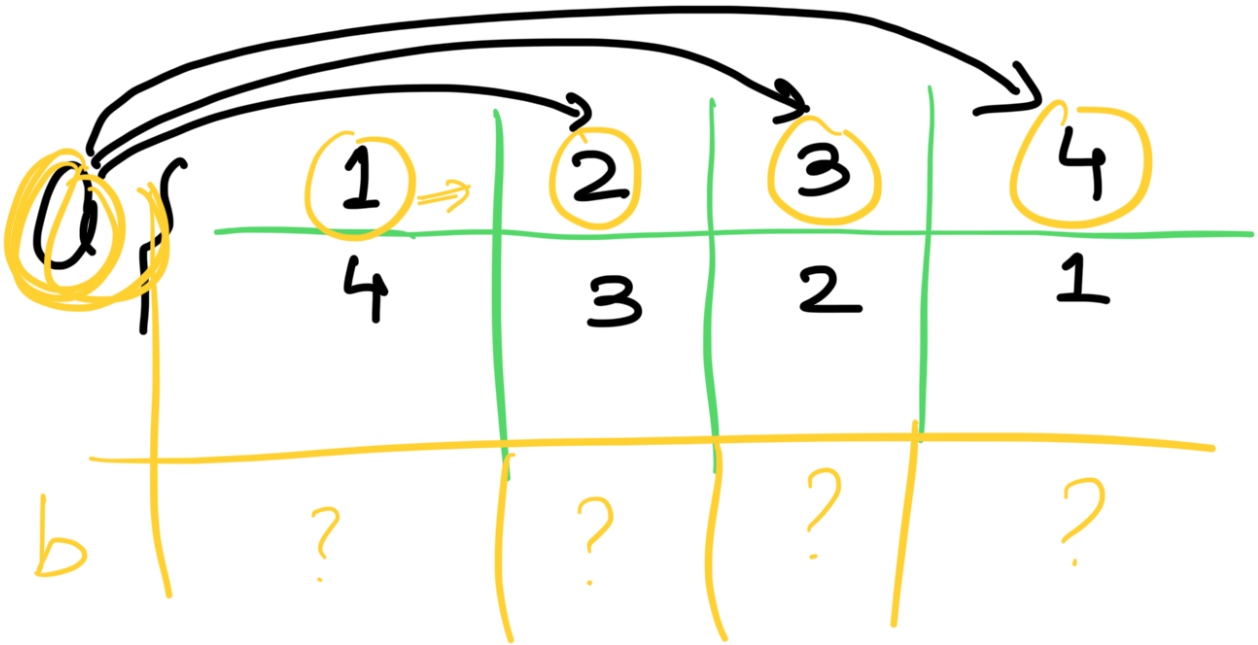
"a" → 1 1+1+1

"aa" → 3

"aa"a → 1+1

"aaaa"

a, 1	4
(a, 2)	3
(a, 3)	2
(a, 4)	1



"aaa" "aaa"
"aa" "aa" "aa"

"a a a b bb aa a

0 1 2 3 4 5 6 7 8

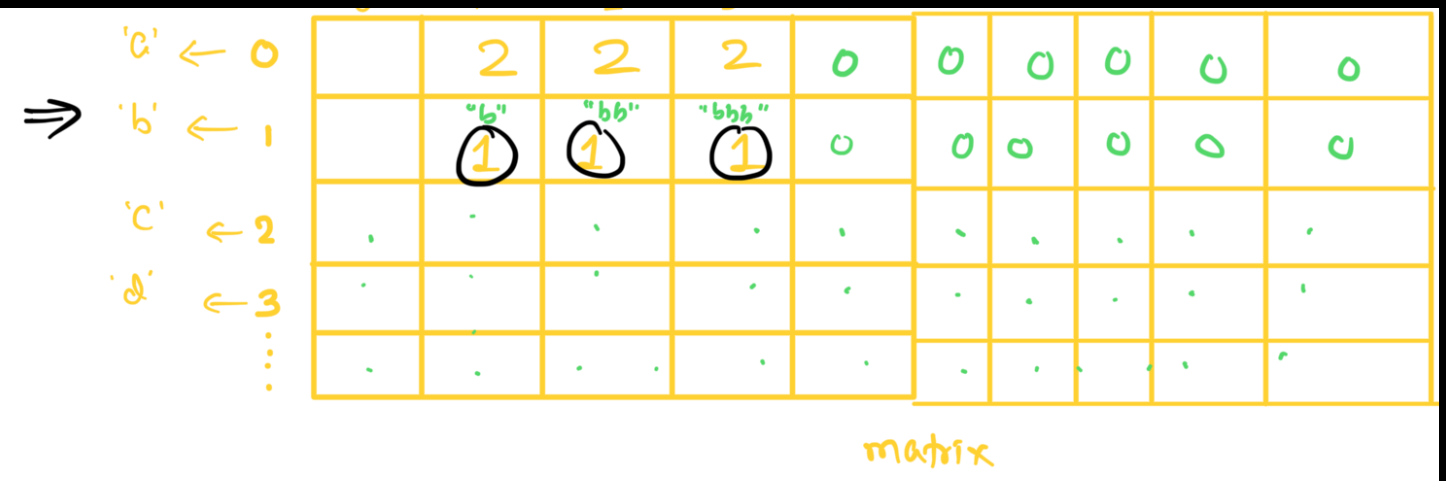
i

n = 9

T.C = O(n)

"a" "aa" "aaa"

0 1 2 3 4 5 6 7 8 9




```
result = -1;
for ( row = 0; row < 26; row++) {
    cumsum = 0;
    for ( int col = n ; col >= 1; col--) {
        cumsum += matrix[row][col];
        if (cumsum >= 3) {
            result = max(result, col);
            break;
        }
    }
```

result = 2

$O(26 * n) \cong O(n)$.

$T.C = O(n)$.