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

STRINGS...

Video-26

Leetcode - 1930

Medium

Unique Length-3 Palindromic Subsequences

Asked By :- Meta amazon Google

1930. Unique Length-3 Palindromic Subsequences

Hint

Medium 805 24

Companies

Given a string `s`, return *the number of* ***unique palindromes of length three*** *that are a* ***subsequence*** *of* `s`.

Note that even if there are multiple ways to obtain the same subsequence, it is still only counted **once**.

A **palindrome** is a string that reads the same forwards and backwards.

A **subsequence** of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

- For example, `"ace"` is a subsequence of `"abcde"`.

Example :-

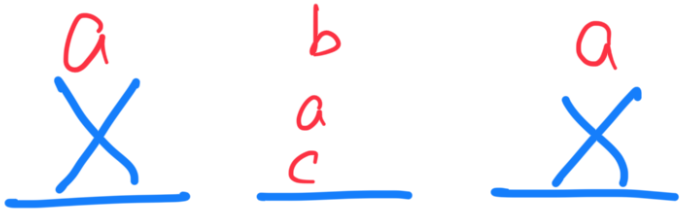
S = "a b a c a"

Output = 3

aba,
aca,
aaa.

Build Intuition :-

"3 length Palindrome"



"a b a c a"
0 1 2 3 4

a → left-idx = 0
↳ right-idx = 4

{b, a, c}

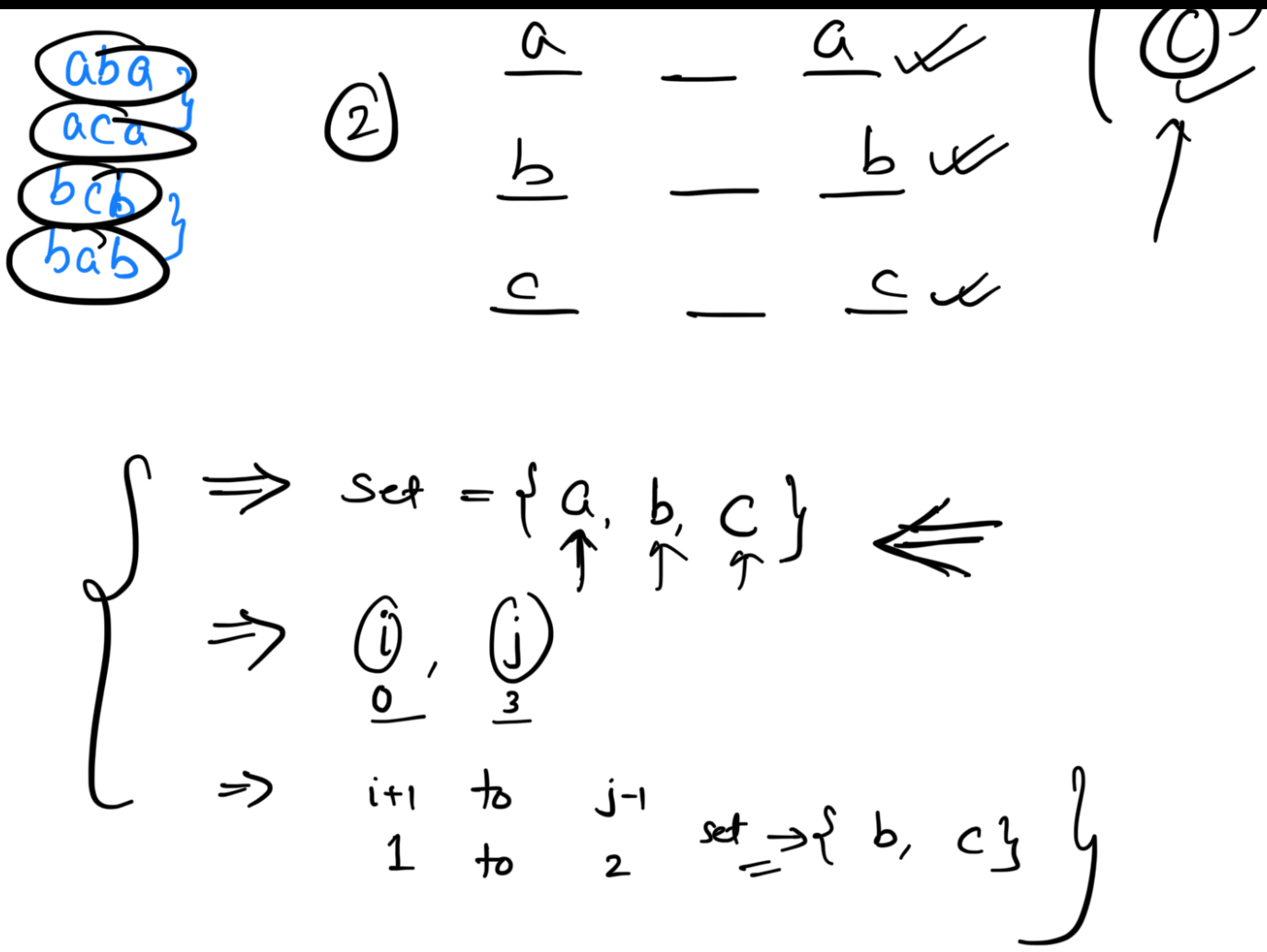
a 1, 2, 3 4
a

"a b a a c a"
0 1 2 3 4 5

0 5
a a

① set<char> → {b, a, c} ← ③
aba aaa aca

S = "a b c a b"
1 2



Time Complexity

```
int result = 0;

for(char letter : letters) {

    //letter = 'a'
    int left_idx  = -1;
    int right_idx = -1;

    for(int i = 0; i<n; i++) {

        if(s[i] == letter) {

            if(left_idx == -1) {
                left_idx = i;
            }

            right_idx = i;
        }

    }

    unordered_set<char> st;
    for(int middle = left_idx+1; middle <= right_idx-1; middle++) {
        st.insert(s[middle]);
    }

    result += st.size();
```

O(26)

precompute left, right idx

space = O(26) = O(1)

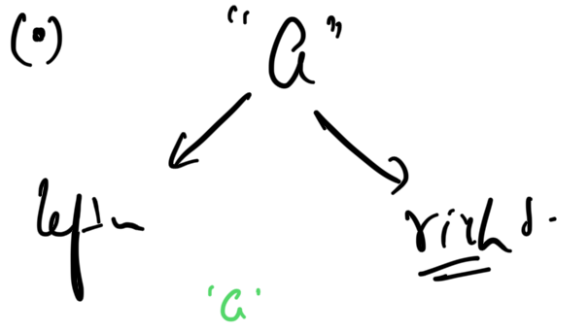
n

O(26*n)

O(n)

n

Approach - 2 (way)



'a' → 0

'b' → 1

{a, b, c}

S = "abcab"

vector<26> { {0, 3}, {1, 4} }