

1358. Number of Substrings Containing All Three Characters

Medium

Topics

Companies

Hint

Given a string `s` consisting only of characters `a`, `b` and `c`.

Return the number of substrings containing **at least** one occurrence of all these characters `a`, `b` and `c`.

Example 1:

Input: `s = "abcabc"`

Output: `10`

Explanation: The substrings containing at least one occurrence of the characters `a`, `b` and `c` are `"abc"`, `"abca"`, `"abcab"`, `"abcabc"`, `"bca"`, `"bcab"`, `"bcabc"`, `"cab"`, `"cabc"` and `"abc"` (again).

Example 2:

Input: `s = "aaacb"`

Output: `3`

```
class Solution {
    public int numberOfSubstrings(String s) {

        int cnt=0 ,n=s.length();
        for(int i=0;i<n;i++){
            int cnta=0 , cntb=0, cntc=0 ;
            for(int j=i;j<n;j++){
                char ch=s.charAt(j);
                switch (ch){
                    case 'a': cnta++; break;
                    case 'b' : cntb++ ; break;
                    case 'c' :cntc++; break;
                }
                if(cnta>0 && cntb>0 && cntc>0){
                    cnt+=(n-j); break;
                }
            }
        }
        return cnt;
    }
}
```

ab cab c

abcabc

abcabc

abcabc

(cnt = n - j)

abcabc

0 1 2 3 4 5

n = 6

j = 2

(cnt = 4)

break.

→ Opti mization

Otherwise

are Counting

abcabc

→

2nd Iteration

abcabc

n = 6

j = 3

cnt = cnt + n - j

= 4 + 6 - 3

= 4 + 3

= 7

... soon ...

TC = O(n²)

Optimal Approach

```
class Solution {
    public int numberOfSubstrings(String s) {
        int[] lastseen={-1,-1,-1};
        int n=s.length() , cnt=0;
        for(int i=0;i<n;i++){
            lastseen[s.charAt(i)-'a']=i;
            int min=Math.min(lastseen[0], Math.min(lastseen[1] , lastseen[2]));
            if(min!=-1){
                cnt+=min+1;
            }
        }
        return cnt;
    }
}
```

→ Concept

→ last occurrence of min(a,b,c) to last of any of them, then (i + 1) ⇒ (idx + 1) ko dekho kote hai.

Ex abcabc

for i = 0

min = -1

→ Do Nothing

for i = 1

min = -1

→ Do nothing

for i = 2

min = 0

cnt = 0 + 1 = 1

ie one substring which contain abc {abc}

for i = 3

min = 1

cnt = 1 + (1 + 1) = 3

→ Basically last element ko base se nikal dekho.

for i = 4

min = 2

cnt = 3 + (min + 1) = 3 + 2 + 1 = 6

new {cab, bcab, abcab} new 3

for i = 5

min = 3

cnt = 6 + (3 + 1) = 6 + 4 = 10

new {abc, cab, bcab, abcabc} new 4.

TC = O(n)

SC = O(1)

This is done By MC - $T_c - O(N)$

```
class Solution {
    public int numberOfSubstrings(String s) {
        int cnta=0 , cntb=0 , cntc=0;
        int cnt=0 , l=0 , r=0 , n=s.length() ;
        while(r<n){
            char ch=s.charAt(r);
            switch (ch){
                case 'a': cnta++; break;
                case 'b' : cntb++ ; break;
                case 'c' : cntc++; break;
            }
            if(cnta>0 && cntb>0 && cntc>0){
                cnt+=(n-r);
            }
            while(cnta!=0 && cntb!=0 && cntc!=0){
                ch=s.charAt(l);
                switch (ch){
                    case 'a': cnta--; break;
                    case 'b' : cntb-- ; break;
                    case 'c' : cntc--; break;
                }
                if(cnta>0 && cntb>0 && cntc>0){
                    cnt+=(n-r);
                }
                l++;
            }
            r++;
        }
        return cnt;
    }
}
```

Concept of Expansion

abc abc

→ Expand until get valid string

i.e

$abcabc$

$\underbrace{\hspace{1.5cm}}_{R}$

$$\Rightarrow \text{cnt} = n - 91 = 6 - 2 = 4 = \left\{ \begin{array}{l} abc \\ abca \\ abcbab \\ abcabc \end{array} \right\}$$

→ Slack until get Not valid

i.e. $abc \quad abc$

$\quad \quad \quad \cup$

$\quad \quad \quad \cup$

↳ nothing

Suppose while shrinking is valid

Ex $\overset{0}{a} \overset{1}{b} \overset{2}{b} \overset{3}{a} \overset{4}{c} \overset{5}{a} \overset{6}{b} \overset{7}{c}$ $len = 8 - 4$
 $= 4$
 abbaca
 abba~~ca~~
 abba~~ca~~
 abba~~ca~~
 abba~~ca~~

→ shrinking

abba**ca**ba

دندلە بە دەستە بەستە

"bbac" is also valid (b)

$$\text{Cnt} = 8 - 4 = \textcircled{4}$$

④ { b b a c
b b a c a
b b a c a b
b b a c a b c }

already is over

$abb \rightarrow \{ \text{left sided tm} \}$

$$T_c = O(N)$$