



Number of substrings containing all three characters

$s = \text{bbacba}$






Number of substrings containing all three characters

s = bbacba





Number of substrings containing all three characters

 s = bbacba

a b c



Number of substrings containing all three characters

$s = \text{bbacba}$

$\text{for } i = 0 \rightarrow n$   
{

$\text{hash}[3] = \{0\}$

$\text{for } j = i \rightarrow n$   
{

$\text{hash}[s[j] - 'a'] = 1$ ;





$s = \text{bbacba}$

```
cnt
fn ( i = 0 → n )
{
    hash[0] = {0}
    fn ( j = i → n )
    {
        hash [ s[j] - 'a' ] = 1 ;
        if ( hash[0] + hash[1] + hash[2] == 3 )
            cnt = cnt + 1 ;
    }
}
```





s = bba**cb**a

7

cnt = 0

for (i = 0 → n)

{  
hash[3] = {0}

for (j = i → n)

{  
hash[s[j] - 'a'] = 1;

if (hash[0] + hash[1] + hash[2] == 3)  
cnt = cnt + 1;

}

}

print(cnt);





s = bbacba

TC  $\rightarrow O(n^2)$

SC  $\rightarrow O(1)$

cnt = 0

for (i = 0  $\rightarrow$  n)

{  
hash[3] = {0}

for (j = i  $\rightarrow$  n)

{  
hash[s[j] - 'a'] = 1;

if (hash[0] + hash[1] + hash[2] == 3)  
cnt = cnt + 1;

}

}

print(cnt);



$S = bba**cb**a$

N 26

$$\tau_c \rightarrow O(w^2)$$
$$SL \rightarrow O(1)$$

0 1 2 3 4 5

cnt = 0

for ( $i = 0 \rightarrow n$ )

$$\ker \{3\} = \{0\}$$
$$f_n(j=i \rightarrow n)$$
$$\text{hakk } \{s[j] - 'a'\} = 1;$$

✓ if (hash[0] + hash[1] + hash[2] == 3)  
    cnt = cnt + 1;

3

3

```
print(cnt);
```









Number of substrings containing all three characters

s = bbacba

With every character, there is a substring that ends.





Number of substrings containing all three characters

S = b b a c b a  
0 1 2 3 4 5

A yellow arrow points to the character 'c' at index 3. A red underline is drawn under the character 'a' at index 2.

With every character, there is a substring that ends.

lastSeen

<sup>2</sup>  
a = ~~-1~~

<sup>1</sup>  
b = ~~-1~~

<sup>3</sup>  
c = ~~-1~~

+ 2





Number of substrings containing all three characters

S = b b a c b a  
0 1 2 3 4 5

With every character, there is a substring that ends.

lastSeen

<sup>5</sup>  
~~2~~  
a = ~~1~~

<sup>4</sup>  
~~2~~  
b = ~~1~~

<sup>3</sup>  
c = ~~1~~

+ 2 + 3



Number of substrings containing all three characters

$S = b b a c b a$

With every character, there is a substring that ends.

lastSeen

$$\begin{array}{r} 5 \\ \cancel{2} \\ a = -1 \end{array}$$
$$b = -1$$
$$C = \cancel{-1}$$
 $+2+3$ 



Number of substrings containing all three characters

S = b b a c b a  
0 1 2 3 4 5

With every character, there is a substring that ends.



Number of substrings containing all three characters

$s = \text{bbacba}$   
0 1 2 3 4 5

with every character, there is a substring that ends.

$f_{\text{unc}}(s)$

$\text{lastseen}[3] = \{-1, -1, -1\};$





func (s)  
{

lastseen[3] = {-1, -1, -1}; cnt = 0

for (i = 0 → n)  
{

lastseen[s[i] - 'a'] = i;

if (lastseen[s







func(s)  
{

lastseen[3] = {-1, -1, -1}; cnt = 0

for (i = 0 → n)

{  
lastseen[s[i] - 'a'] = i;

if (lastseen[0] != -1 && ... )  
cnt = cnt + (1 + min(lastseen[0], s[1], s[2]))





func(s)  
{

lastseen[3] = {-1, -1, -1}; cnt = 0

for (i = 0 → n)

{  
lastseen[s[i] - 'a'] = i;

→ if (lastseen[0] != -1 && ... )  
cnt = cnt + (1 + min(lastseen[0], s[1], s[2]))

}

return cnt;

}





func(s)  
{

lastseen[3] = {-1, -1, -1}; cnt = 0

for (i = 0 → n)  
{

TC → O(N)

SC → O(1)

lastseen[s[i] - 'a'] = i;

→ if (lastseen[0] != -1) ...  
cnt = cnt + (1 + min(lastseen[0], s[1], s[2]))

{  
return cnt;  
}

1 - 1  
0

↗  
-1





func(s)  
{

lastseen[s] = {-1, -1, -1}; cnt = 0

for(i = 0 → n)  
{

lastseen[s[i] - 'a'] = i;

TC → O(N)

SC → O(1)

→ if (lastseen[s[i]] != -1) {  
cnt = cnt + (1 + min(lastseen[s[i]], s[i], s[i+1]))

}  
return cnt;  
}

1 - 1  
0

↗  
-1

