

Number of subsequences of the form $a^i b^j c^k$

Given a string, count number of subsequences of the form $a^i b^j c^k$, i.e., it consists of i 'a' characters, followed by j 'b' characters, followed by k 'c' characters where $i \geq 1$, $j \geq 1$ and $k \geq 1$.

Note: Two subsequences are considered different if the set of array indexes picked for the 2 subsequences are different.

Expected Time Complexity : $O(n)$

Examples:

```
Input  : abbc
Output : 3
Subsequences are abc, abc and abbc

Input  : abcabc
Output : 7
Subsequences are abc, abc, abbc, aabc
abcc, abc and abc
```

Asked in : Amazon

Recommended: Please solve it on “**PRACTICE**” first, before moving on to the solution.

We traverse given string. For every character encounter, we do following:

- 1) Initialize counts of different subsequences caused by different combination of 'a'. Let this count be aCount.
- 2) Initialize counts of different subsequences caused by different combination of 'b'. Let this count be bCount.
- 3) Initialize counts of different subsequences caused by different combination of 'c'. Let this count be cCount.
- 4) Traverse all characters of given string. Do following for current character **s[i]**

If current character is 'a', then there are following possibilities :

- a) Current character begins a new subsequence.
- b) Current character is part of aCount subsequences.
- c) Current character is not part of aCount subsequences.

Therefore we do $aCount = (1 + 2 * aCount)$;

If current character is 'b', then there are following possibilities :

- a) Current character begins a new subsequence of b's with aCount subsequences.
- b) Current character is part of bCount subsequences.
- c) Current character is not part of bCount subsequences.

Therefore we do $bCount = (aCount + 2 * bCount)$;

If current character is 'c', then there are following possibilities :

- a) Current character begins a new subsequence of c's with bCount subsequences.
- b) Current character is part of cCount subsequences.
- c) Current character is not part of cCount subsequences.

Therefore we do $cCount = (bCount + 2 * cCount)$;

5) Finally we return cCount;

Below is C++ implementation of the idea.

```
// C++ program to count subsequences of the
// form  $a^i b^j c^k$ 
#include <bits/stdc++.h>
using namespace std;

// Returns count of subsequences of the form
//  $a^i b^j c^k$ 
int countSubsequences(string s)
{
    // Initialize counts of different subsequences
    // caused by different combination of 'a'
    int aCount = 0;

    // Initialize counts of different subsequences
    // caused by different combination of 'a' and
    // different combination of 'b'
    int bCount = 0;

    // Initialize counts of different subsequences
    // caused by different combination of 'a', 'b'
    // and 'c'.
    int cCount = 0;

    // Traverse all characters of given string
    for (unsigned int i=0; i<s.size(); i++)
    {
        /* If current character is 'a', then
        there are following possibilities :
        a) Current character begins a new
        subsequence.
        b) Current character is part of aCount
        subsequences.
        c) Current character is not part of
        aCount subsequences. */
        if (s[i] == 'a')
            aCount = (1 + 2 * aCount);

        /* If current character is 'b', then
        there are following possibilities :
        a) Current character begins a new
        subsequence of b's with aCount
        subsequences.
        b) Current character is part of bCount
        subsequences.
        c) Current character is not part of
        bCount subsequences. */
        else if (s[i] == 'b')
            bCount = (aCount + 2 * bCount);

        /* If current character is 'c', then
        there are following possibilities :
        a) Current character begins a new
        subsequence of c's with bCount
        subsequences.
        b) Current character is part of cCount
```

```
        subsequences.  
        c) Current character is not part of  
        cCount subsequences. */  
    else if (s[i] == 'c')  
        cCount = (bCount + 2 * cCount);  
    }  
  
    return cCount;  
}  
  
// Driver code  
int main()  
{  
    string s = "abbc";  
    cout << countSubsequences(s) << endl;  
    return 0;  
}
```

[Run on IDE](#)

Output:

3

Time Complexity : $O(n)$

This article is contributed by **Mr. Somesh Awasthi**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.