

Analysis: Welcome to Code Jam

In this problem, every one is welcomed by a (somehow) standard dynamic programming problem.

The word we want to find is $S = \text{"welcome to code jam"}$, in a long string T . In fact the solution is not very different when we want to find any S . It is actually illustrative to picture the cases for short words.

In case S is just a single character, you just need to count how many times this character appears in T . If $S = \text{"xy"}$ is a string of length 2, instead of brute force all the possible positions, one can do it in linear time, start from left to the right. For each occurrence of 'y', one needs to know how many 'x's appeared before that 'y'.

The general solution follows this pattern. Let us again use $S = \text{"welcome to code jam"}$ as an example. The formal solution will be clear from the example; and you can always download the good solutions (with nice programming techniques) from the scoreboard.

So, let us define, for each position i in T , $T^{(i)}$ to be the string consists of the first i characters of T . And write

- $Dp[i,1]$: How many times we can find "w" in $T^{(i)}$?
- $Dp[i,2]$: How many times we can find "we" in $T^{(i)}$?
- $Dp[i,3]$: How many times we can find "wel" in $T^{(i)}$?
- $Dp[i,4]$: How many times we can find "welc" in $T^{(i)}$?
- ...
- $Dp[i,18]$: How many times we can find "welcome to code ja" in $T^{(i)}$?
- $Dp[i,19]$: How many times we can find "welcome to code jam" in $T^{(i)}$?

Assume $Dp[i,j]$ is computed for each j , let us see how easy we can compute, say, $Dp[i+1,4]$:

1. If the $(i+1)$ -th character of T is not 'c', then $Dp[i+1,4] = Dp[i,4]$.
2. If the $(i+1)$ -th character of T is 'c', then we can include all the "welc"s found in $T^{(i)}$, as well as those "welc"s ends exactly on the $(i+1)$ -th character, so $Dp[i+1,4] = Dp[i,4] + Dp[i,3]$.

Finally, let n be the length of the text T , $Dp[n,19]$ will be our answer.

That's it. Welcome to Code Jam; and we hoped you enjoyed this round.