Distinct Subsequence

Question: Given a string S and a string T, count the number of distinct subsequences of T in S.

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, "ACE" is a subsequence of "ABCDE" while "AEC" is not).

Here is an example:

S = "rabbbit", T = "rabbit"

Return 3.

Solutions:

```
class Solution:
    # @return an integer
    # @dp
    # dp[i][j] means how many first j of T is sub of first i of S.
    def numDistinct(S, T):
        dp = [[0 for i in range(len(T)+1)] for j in range(len(S)+1)]
        for j in range(len(S)+1):
            dp[j][0] = 1
        for i in range(1, len(S)+1):
            for j in range(1, min(i+1, len(T)+1)):
                if S[i-1] == T[j-1]:
                    dp[i][j] = dp[i-1][j] + dp[i-1][j-1]
                else:
```

```
                dp[i][j] = dp[i-1][j]
        return dp[len(S)][len(T)]


Solution.numDistinct("rabbbit","rabbit")
```