DP32: [Distinct Subsequences](https://leetcode.com/problems/distinct-subsequences/)

**Approach**: we have two strings we have to count how many str2 subsequences we can make out of str1. We can use something similar to LCS

=> We will start matching from the n-1, m-1

1. If str1[i] == str2[j] matches; we have two possibilities

* Accept that this i we will consider and now search in rest i-1 & j-1
* We want to search another jth char in str1 so shrink i -> i-1 and j as it is.

1. If they doesn't match, shrink i -> i-1 to match it with j

**Recursion**:

1. f(i, j)
2. Match- not match
3. Return total number of distinct subseq.

**if(s[i]==t[j])**

**return f(i-1, j-1, s, t) + f(i-1, j, s, t);**

**//if its not matching, you definitely have to search inside in s1 for jth char**

**else**

**return f(i-1, j, s, t);**

**Tabulation:**

1. Right shift of index to deal with i<0 and j<0 base cases: n->n+1, m->m+1
2. Use base case to make the base of the table
3. Use recurrence relation to form table i=1 to n, j=1 to m