

115. Distinct Subsequences

题目描述: <https://leetcode.com/problems/distinct-subsequences/>

给定两个字符串s,t判断s有几种subsequence可以==t

例如:

```
s = "rabbbit" t = "rabbit"
return 3
```

解题思路:

dp[i][j] 代表s的前i位变成t的前j位有多少种subsequence。

则初始化: $0 \leq i \leq \text{lenS}$ dp[i][0] = 1;

dp[i][j] = dp[i-1][j-1] + dp[i-1][j], s[i-1] == t[j-1] //如果s[i-1]和t[j-1]相等则可以选择用i这一位, 或者不用

dp[i][j] = dp[i-1][j], s[i-1] != t[j-1] //只能不用

代码:

```
class Solution {
public:
    int numDistinct(string s, string t) {
        int l1 = s.size(), l2 = t.size();
        vector<vector<int>> > f(l1+1, vector<int>(l2+1));
        for(int i = 0; i <= l1; i++) {
            f[i][0] = 1;
        }
        for(int i = 1; i <= l1; i++) {
            for(int j = 1; j <= l2; j++) {
                if(s[i-1] == t[j-1]) {
                    f[i][j] = f[i-1][j-1] + f[i-1][j];
                }
                else {
                    f[i][j] = f[i-1][j];
                }
            }
        }
        return f[l1][l2];
    }
};
```

