97. Interleaving String

题目描述:https://leetcode.com/problems/interleaving-string/

> 给定三个字符串s1.s2.s3，判断s3是不是由s2插入s1形成的
> 例如:
>
> ```
> s1 = "abc" s2 = "ddd" s3 = "adbcdd" ---> true
> s1 = "abc" s2 = "ddd" s3 = "adddbc" ---> true
> s1 = "abc" s2 = "ded" s3 = "adbcde" ---> false
> ```

解题思路:

f[i][j]代表由s1前i位 s2的前j位 能否组成 s3的前i+j位
f[i][j] = (f[i][j-1] && s2[j-1]== s3[i+j-1]) || (f[i-1][j] && s1[i-1] && s3[i+j-1])

代码:

```
class Solution {
public:
    bool isInterleave(string s1, string s2, string s3) {
        int len1 = s1.size(), len2 = s2.size(), len3 = s3.size();
        if(len1+len2 != len3) return false;
        vector<vector<int> > f(len1+1, vector<int>(len2+1));
        for(int i = 0; i <= len1; i++) {
            for(int j = 0; j <= len2; j++) {
                if(i == 0 && j == 0) {
                    f[i][j] = true;
                }
                else if(i == 0 && j != 0) {
                    f[i][j] = f[i][j-1] && (s2[j-1] == s3[i+j-1]);
                }
                else if(j == 0 && i != 0) {
                    f[i][j] = f[i-1][j] && (s1[i-1] == s3[i+j-1]);
                }
                else {
                    f[i][j] = (f[i][j-1] && (s2[j-1] == s3[i+j-1]))||(f[i-1][j] && (s1[i-1] == s3[i+j-1]));
                }
            }
        }
        return f[len1][len2];
    }
};
```