一、题目说明

这个题目是10. Regular Expression Matching，乍一看不是很难。

但我实现提交后，总是报错。不得已查看了答案。

二、我的做法

我的实现，最大的问题在于对`.*`的处理有问题，始终无法成功。

```
#include<iostream>

using namespace std;

class Solution{
    public:
    bool isMatch(string s,string p){
        bool result = true;
        if(s.length()<=0 && p.length()<=0){
            return true;
        }
        if(p==".*"){
            return true;
        }

        int sCurr=0,pCurr=0;
        int lenS = s.length();
        int lenP = p.length();

        //count the num of .*
        int numOfWildCard = 0;
        while(pCurr<lenP){
            if(p[pCurr]=='.' && pCurr+1<lenP && p[pCurr+1]=='*'){
                numOfWildCard++;
            }

            pCurr++;
        }
        //cout<<numOfWildCard<<":";

        pCurr = 0;
        while(sCurr<lenS && pCurr<lenP){
            if((pCurr+1<lenP) && p[pCurr]=='.' && p[pCurr+1]=='*'){
                if(pCurr+2<lenP){
                    pCurr = pCurr+2;
                    while(sCurr<lenS && s[sCurr]!=p[pCurr]){
                        sCurr++;
                    }
                }

            }
            if((pCurr+1<lenP) && p[pCurr+1]=='*'){
                while(sCurr<lenS && s[sCurr]==p[pCurr]){
                    sCurr++;
                }
                pCurr = pCurr+2;
```

```
            }
            if(sCurr<lenS && pCurr<lenP && p[pCurr+1]!='*'){
                if(s[sCurr]==p[pCurr] || p[pCurr]=='.'){
                    sCurr++;
                    pCurr++;
                }
            }
        }

        if(sCurr==lenS && pCurr==lenP){
            result = true;
        }else{
            result = false;
        }
        return result;
    }
};


int main(){
    Solution s;
    cout<<(false==s.isMatch("aa","a"))<<endl;
    cout<<(true==s.isMatch("aa","a*"))<<endl;
    cout<<(true==s.isMatch("ab",".*"))<<endl;
    cout<<(true==s.isMatch("aab","c*a*b"))<<endl;
    cout<<(false==s.isMatch("mississippi","mis*is*p*."))<<endl;
    return 0;
}
```

三、正确的做法

1.递归方法

```
#include<iostream>
#include<vector>

using namespace std;

class Solution{
    public:
    bool isMatch(string s, string p) {//aa a
        if(p.empty()) return s.empty();
        if(s.empty()) return p.empty() || (p[1] == '*' ? isMatch(s, p.substr(2))
: false);
        if(p[0] != '.' && s[0] != p[0]) return p[1] == '*' ? isMatch(s,
p.substr(2)) : false;
        if(p[1] == '*') return isMatch(s.substr(1), p) || isMatch(s,
p.substr(2));
        return isMatch(s.substr(1), p.substr(1));
    }
};

int main(){
    Solution s;
    cout<<(false==s.isMatch("aa","a"))<<endl;
    cout<<(true==s.isMatch("aa","a*"))<<endl;
```

```
    cout<<(true==s.isMatch("ab",".*"))<<endl;
    cout<<(true==s.isMatch("aab","c*a*b"))<<endl;
    cout<<(false==s.isMatch("mississippi","mis*is*p*."))<<endl;
    cout<<(false==s.isMatch("ab",".*c"))<<endl;
    cout<<(true==s.isMatch("aaa","a*a"))<<endl;
    return 0;
}
```

2.DP方法

dp是什么？动态规划啊，

```
#include<iostream>
#include<vector>
#include <mem.h>

using namespace std;

class Solution {
public:
    bool isMatch(string s, string p) {
        int ssize = s.size(),psize = p.size();
        string pp="";
        vector<bool> star;

        for(int i=0;i<p.size();i++){
            if(p[i]=='*'){
                star.back()=true;
            }else{
                star.push_back(false);
                pp+= p[i];
            }
        }

        psize = pp.size();

        bool dp[psize+1][ssize+1];
        memset(dp,false,sizeof(dp));

        dp[0][0] = true;
        for(int i=1;i<=psize;i++){
            if(star[i-1]==true){
                dp[i][0] = true;
            }else{
                break;
            }
        }

        for(int i=1;i<=psize;i++)
            for(int j=1;j<=ssize;j++){
                if(dp[i-1][j-1]== true){
                    if(pp[i-1]==s[j-1] || pp[i-1]=='.'){
                        dp[i][j] = true;
                        continue;
                    }
                }
```

```
                if(dp[i-1][j]== true){
                    if(star[i-1]==true){
                        dp[i][j] = true;
                        continue;
                    }
                }

                if(dp[i][j-1]== true){
                    if(star[i-1]==true && (pp[i-1]==s[j-1] || pp[i-1]=='.')){
                        dp[i][j] = true;
                        continue;
                    }
                }
            }
        return dp[psize][ssize];
    }
};

int main(){
    Solution s;
    cout<<(false==s.isMatch("aa","a"))<<endl;
    cout<<(true==s.isMatch("aa","a*"))<<endl;
    cout<<(true==s.isMatch("ab",".*"))<<endl;
    cout<<(true==s.isMatch("aab","c*a*b"))<<endl;
    cout<<(false==s.isMatch("mississippi","mis*is*p*."))<<endl;
    cout<<(false==s.isMatch("ab",".*c"))<<endl;
    cout<<(true==s.isMatch("aaa","a*a"))<<endl;
    cout<<(false==s.isMatch("a",""))<<endl;
    return 0;
}
```

四、总结

看来基础知识还需要恶补，加油!