B 题解题报告

【题目描述】

A 和 B 分别有 n 和 m 个糖果，他们玩一个游戏，游戏每一轮解一道题目,A 和 B 解题成功率分别是 p 和 q，如果两人都解出或都没有解出算平局，否则败者要给胜者一颗糖，知道一方糖果数为 0，另一方胜利，游戏结束。

【思路】

递推。

设 dp[i]为 A 手上拥有 i 个糖果的胜利概率.

很容易得到转移方程

$$dp[i]=p(1-q)dp[i-1]+q(1-p)dp[i+1]+(1-p-q+2pq)dp[i]$$

整理后可以得到一个递推式

$$dp[i+1]=\frac{p+q-2pq}{p(1-q)}dp[i]-\frac{q(1-p)}{p(1-q)}dp[i-1]$$

令 $x=\frac{p+q-2pq}{p(1-q)}$ ， $y=\frac{q(1-p)}{p(1-q)}$

其中 dp[0]=0,dp[n+m]=1

式子的初始状态不好确定,因为 dp[1]未知。

我们不妨将 dp[1]看做一个未知数。

即

$$dp[2]=xdp[1]-ydp[0]=xdp[1]$$

$$dp[3]=xdp[2]-ydp[1]=(x^2-xy)dp[1]$$

......

$$dp[n+m]=f(x,y)dp[1]$$

又因为 dp[n+m]=1,所以我们可以算出 dp[1]

所以递推时我们可以令 dp[1]=1

那么最后求得的结果就是 $\frac{dp[n]}{\mathrm{dp}[n+m]}$

另外，注意处理好几个特殊情况，n=0，m=0，p=0 || q=1

【code】

```
# include <cstring>
# include <cstdio>
# include <cstdlib>
# include <iostream>
using namespace std;
double dp[200];
int main (void)
{
    int m,n;
    double p,q;
```

```
    while (cin>>n>>m>>p>>q)
    {
        if (n==0)
            cout<<"0.00"<<endl;
        else
            if (m==0)
                cout<<"1.00"<<endl;
        else
            if (p==0.0 || q==1.0)
                cout<<"0.00"<<endl;
        else
        {
            double x=(p+q-2*p*q)/(p*(1-q));
            double y=q*(1-p)/(p*(1-q));
            dp[0]=0;
            dp[1]=1;
            for (int i=2;i<=n+m;i++)
                dp[i]=dp[i-1]*x-dp[i-2]*y;
            printf("%.2f\n",dp[n]/dp[n+m]);
        }
    }
    return 0;
}
```