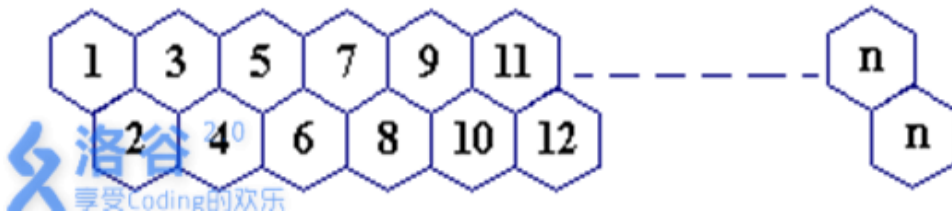


P2437 蜜蜂路线

题目描述

一只蜜蜂在下图所示的数字蜂房上爬动, 已知它只能从标号小的蜂房爬到标号大的相邻蜂房, 现在问你: 蜜蜂从蜂房 m 开始爬到蜂房 n , $m < n$, 有多少种爬行路线? (备注: 题面有误, 右上角应为 $n-1$)



输入格式

输入 m, n 的值

输出格式

爬行有多少种路线

输入样例

1 14

输出样例

377

解析

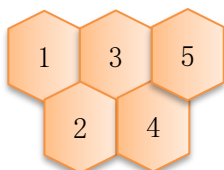
这道题与上楼梯非常类似, 本质都是每一次只能上1层或者两层, 如下图所示:

到达3层的方法有两个, 直接从1号过去, 或者从2号过去。

到达4层的方法有两个, 直接从3号过去, 或者从2号过去。

因此, $f(n) = f(n-1) + f(n-2)$

但是本题的 n 的数量非常巨大, 因此, 需要使用高精度加法。



编码

```
#include <bits/stdc++.h>

using namespace std;
string result[1001] = {"0", "1", "1"};
int n, m;

string countSum(string numA, string numB) {
    int lenA = numA.size();
    int lenB = numB.size();
    string ans;
    //105位数字，是一个极大的数字
    int a[1000] = {0}, b[1000] = {0};
    //存储真正读入的数字，逆序存储
    for (int i = 0; i < lenA; i++) {
        a[i] = numA[lenA - 1 - i] - '0';
    }
    for (int i = 0; i < lenB; i++) {
        b[i] = numB[lenB - 1 - i] - '0';
    }
    lenA = max(lenA, lenB); //保证后续的计算按照位数最高的进行循环。
    for (int i = 0; i < lenA; i++) {
        a[i] += b[i]; //按位进行加法运算
    }
    for (int i = 0; i < lenA; i++) {
        a[i + 1] += a[i] / 10; //高位进位
        a[i] %= 10; //低位保留余数
    }
    while (a[lenA]) { //循环判断最高位是不是需要进行进位
        a[lenA + 1] += a[lenA] / 10;
        a[lenA] %= 10;
        lenA++;
    }
    //记录最终结果
    for (int i = lenA - 1; i >= 0; i--) {
        //将单个数字还原为字符
        ans += a[i] + '0';
    }
    return ans;
}

int main(int argc, char **argv) {
    cin >> m >> n;
    //无论从多少开始，起始的第一步都是1，第二步都是2
    int res = n - m + 1;
    for (int i = 3; i <= res; i++) {
        result[i] = countSum(result[i - 1], result[i - 2]);
    }
}
```

```
cout << result[res];  
return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

