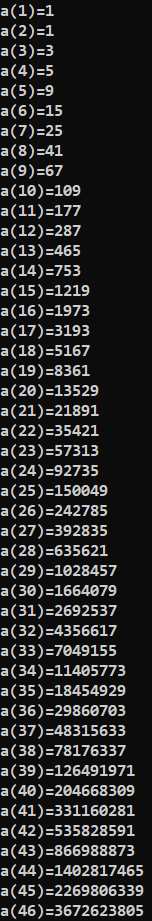
递归法求 Fibonacci 数列中的递归次数分析

相关的函数如下：

int fibonacci(int n)  
{  
 if (n == 1 || n == 2)  
 return 1;  
 else  
 return fibonacci(n - 1) + fibonacci(n - 2);  
}

设输入为 时，函数调用了 次。有初始条件：

对于 较大的情况，首先调用一次，接着再依次调用 和 . 得到递推公式：

可以构造程序计算不同输入对应的递归次数

#include <iostream>  
using namespace std;  
long long recursion\_times(int n)  
{  
 if (n == 1 || n == 2)  
 return 1;  
 else  
 return recursion\_times(n - 1) + recursion\_times(n - 2) + 1  
}  
int main()  
{  
 int n = 1;  
 for (; n <= 46; ++n)  
 {  
 cout << "a(" << n << ")=" << recursion\_times(n) << endl;  
 }  
 return 0;  
}  
结果如右图。计算一个 Fibonacci 数列中的数所需要进行的递归次数比这个数本身还要大。