递归问题思路

1.小乐乐爬楼梯，一次可爬1或2阶，输入台阶数n（1~30），输出小乐乐爬楼的种数。

这种问题一般从两端分析：

如果从第一步开始分析，小乐乐第一步只能走一步，或者两步，假设存在一个函数f（n），其功能是求取当台阶数为n的时候的爬楼方法数，则f（n）即代表总方法数，如果先走一步，那么方法数为f（n-1），如果先走两步，那么方法数为f（n-2），而不管第一步怎么走，这两种情况均属于总方法数，即：

f（n）=f（n-1）+f（n-2）

显然，由n>2，对于n=1，方法数为1；对于n=2，方法数为2；

如果从最后一步进行分析，亦是同理，最后一步只能是1或2步，那么也就是f（n-1）和f（n-2），两种情况加和，其他思路与前者一致。下面是实现代码：

#include<stdlib.h>

#include<time.h>

int test(int n)

{

if (n > 2)

{

return test(n-1) + test(n-2);

}

else

{

return n;

}

}

int main()

{

int n = 0;

int i = 1;

do

{

printf("n=");

scanf("%d", &n);

if (n > 0 && n <= 30)

{

i = 0;

}

} while (i);

test(n);

printf("方法数=%d\n", test(n));

return 0;

}

2.