# 递归函数总结

## 递归函数的终止

今天在书写Hanoi问题的函数时，我发现加上if和不加if的情况下程序执行完全相反：不加上if程序无法执行。

显然，这个问题是如何终止递归函数的执行（即告诉编译器不要再递归下去）？

回忆Factorial函数和Fibonacci函数，这两个函数都有边界条件。

递归函数的边界条件是当递归函数的参数满足边界条件的时候，函数无需再继续递归下去，而是去执行满足边界条件时的代码段。（显然当递归函数满足边界条件的时候，递归函数就无需再继续往下递归了）

因此加上边界条件能够告诉编译器不要继续往下递归，从而让程序回溯（也就是返回去执行上层的函数）。

现在在来考虑Hanoi问题的程序，我已经发现了问题的症结所在：如果不加if (n > 0)，那么程序将没有递归的终止条件，理论上将没有止境地递归下去。

## 递归和分治

分治的思想是：对于难以直接求解的问题，可以考虑把这个问题分解成若干规模较小的相同子问题，从而各个击破，分而治之。

分治法采用递归函数实现的原因：由分治法产生的子问题和原问题往往是相同性质的，因此可以使用递归技术。而分治法的核心是把问题的规模降低，因此，在回溯函数的参数中往往有表示问题规模的参数。