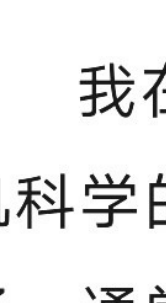


# 第118封信 | 再谈递归原理



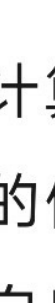
吴军

今天 00:00



第118封信 | 再谈递归原理

09:02 4.23MB



信件朗读者：宝木

## 小师弟，你好！

我在之前的来信中讲过，递归是计算机科学的精髓，并且讲述了一些递归的例子。通常人的思维方式是从小到大，自底向上的递推思维方式，但是**计算机则相反，它是自顶向下，从整体到局部的递归思维**。但是，什么是递归实在很难讲清楚，以至于大约有三分之一的计算机从业者并不能很好地使用递归来设计程序。因此，我估计你上次听八皇后的内容也是一知半解，这不是你的问题，是我的问题。

这段时间，我一直琢磨用什么例子给你把递归讲解清楚。几天前我和Google的一个同事吃饭，他讲了最近面试一个人的情况，我发现他出的那道面试题是个非常好的例子，今天我们就用他的例子再来谈谈递归。

这道面试题说难也难，说容易也容易，可能有的小学生也能做出来。题目是这么说的：

我们俩来做一个游戏。第一个人先从1和2中挑一个数字，第二个人可以在对方的基础上选择加1，或者加2。然后又轮到了第一个人，他可以再次选择加1，或者加2，之后把选择权交给对方。就这样双方交替地选择加1或者加2，谁要是正好加到20，谁就赢了。用什么策略保证一定能赢？

为了方便你理解，我们举一个实际的例子，当然加到20的时间太长，我们假设加到10。假如我让你先选，你选了2。接下来我选择加2，这样我加到了4，然后你选择加1，你加到了5，这时我还会选择加2，就加到了7。接下来，不论你选择加1到8，还是加2到9，我都能加到10，因此我赢定了。当然，你可能觉得先作选择的人吃亏，那么这次我先来。我选择1，你可能会选择加2到3，然后我选择加1到4。接下来，假如你选择加2到6，我会选择加1到7，这又回到第一次最后的状态了，我还是赢了。

可能你已经从上面的例子中想清楚这道题里面的技巧了，如果你还没有想清楚，我再等你五秒钟……好了，我们回来讲讲这道题。其实如果仅仅是抢10，情况并不复杂，你即使想不清楚它的道理，试几次也能找到规律，但是如果是抢20，情况就复杂多了。如果是抢30，抢50呢？就不能通过穷举法这种笨办法解决问题了，就必须找它的规律了。

可能你已经看出来了。要想抢到20，就需要抢到17，因为抢到了17，无论对方是加1还是加2，你都可以加到20。而要想抢到17，就要抢到14，以此类推，就必须抢到11、8、5和2。因此对于这道题，只要第一个人抢到了2，他就赢定了。这里面最核心的地方在于看清楚，**无论对方选择1还是2，你都可以让这一轮两个人加起来的数值等于3**，于是你就可以牢牢控制整个过程了。

那么这道看似是智力题的面试题是要考察候选人的什么技能呢？就是对计算机递归思想的理解。对于一般的人，让他们数数，数到20。他们会从小往大数，但是这道题的解题思想正好相反。它是要寻找20，就要先寻找17，至于怎么从17到20，方法你是知道的。接下来要寻找17，就要寻找14，等等，这就是递归的思想。

上述思想其实在我们做事情的时候也经常采用。比如我要完成一本书的出版，有两个办法，一个办法是顺序估算每一个任务的时间。首先我要在一个月内交稿，出版社要在一个月内完成一校，然后我在某个时间完成修改，最后在某月某日出版，这是一种做法。

还有一种做法是倒推，或者倒逼。比如我们要在“双十一”前上市销售，那么书就必须在11月1日入库，在10月15日开印，在10月5日定稿，等等，然后倒推我交稿的时间，一校、二校完成的时间等等。

哪种工作方式更有效呢？通常是后一种。另外，如果按照后一种方式工作，你会发现很多事情根本不可能在规定的时间内做完，那么怎么办呢？办法很简单，就是做减法。你可以认为，上面这种工作方法，其实和计算机思维中的递归在原理上是一致的。

上面这道面试题，可能有点过于简单，但是面试官其实还留有了两个后手。

首先，他会问面试者，按照上述方法，从一开始（以1为起点）加到20，有多少种不同的递加过程？比如1，4，7，10，12，15，18，20是一种；2，5，8，11，14，17，20又是一种。那么这样的过程有多少种呢？这道题显然不简单了吧？下面我再给你5秒钟想一下。

好了，5秒钟时间过去了，我估计绝大部分人还是想不出来。事实上，面试Google的人大部分在两分钟内根本想不出这道题的答案，更不要说5秒钟了。因此你如果没有想出来，不要气馁，接下来我给你一讲就明白了。

解这道题的技巧也在于使用递归，如果你从1，2，3开始找规律就难了。我们假定数到20有F（20）种不同的路径，那么到达20这个数字，前一步只有两个可能的情况，即从18直接蹦到20，或者从19数到20。

由于从18蹦到20，和19到20，彼此是不同的，因此走到20的路径数量，其实就是走到18的路径数量，加上走到19的路径数量，也就是说 $F(20) = F(18) + F(19)$ 。类似地， $F(19) = F(18) + F(17)$ 。这就是递推公式。

最后，F（1）只有一个可能性，就是1，F（2）有两个可能性，要么直接蹦到2，要么从1走到2。知道了F（1）和F（2），就可以知道F（3），然后再倒着推导回去，一直到F（20）即可。

数学比较好的朋友，可能已经看出来这是著名的斐波那契数列，如果我们认为F（0）也等于1，那么这个数列是这样的，1（=F（0）），1，2，3，5，8，13，21，……这个数列几乎按照几何级数的速度增长，到了F（20），就已经是10946了。因此，靠穷举法是不可能把所有情况想清楚的。

**在数学和计算机上，还有一个非常重要的原则，就是等价性原则，也就是说很多问题都是等价的。**比如说，我再给你出一道题，如果一个楼梯有20层，你每次可以走一层或者两层，爬到20层有多少种走法？这个问题的解，和抢20是一样的，也是斐波那契数列。

今天，我通过一个不算太复杂的问题，再一次讲解了递归的思想，你把它理解为生活和工作中的倒逼就好了。另外，我再一次强调了**计算机科学和数学中的等价性原则，掌握了这个原则，就可以把很多问题归结为一类问题，解决了其中的一个，其他的就迎刃而解。**

今天给大家留两道思考题：

1. 抢40的策略，每个人每次可以选择加1到4中的任何一个数字，如果你先开始，你怎样能赢？
2. 能否谈谈你对工作中对倒逼的方法的体会？

祝近安



2018年6月18日

——寻找知识伙伴——

今天的思考题中的第一个，抢40游戏，你可以通过“请朋友读”寻找你的一位知识伙伴，一起讨论。

为了鼓励你表达，以及可以发起有碰撞的讨论，我们会在**6月16日—6月22日（7日内）找到分享来信最多次的一位读者，送出一份《吴军的谷歌方法论》的兑换码**，让你的知识伙伴，可以和你一起读一整年的来信。

你可以点击每篇内容右下角的“请朋友读”分享，现在就开始寻找一位你的知识伙伴吧！

**吴军的谷歌方法论**  
一份智能时代的行动指南  
版权归得到App所有，未经许可不得转载



Aa



写留言



88



请朋友读