NOIP 2003 普及组 No.3《栈》解题报告

山西省实验中学 郝嘉
大榕树编程世界 http://drs.126.com

收稿日期：2003 年 12 月 26 日

编者按：

本文十分详细的介绍了《栈的计数》这题的两种递推算法，图文并茂，十分适合初学者阅读，并从中领会清晰、严密的分析思路。另外，随文附带的程序十分简洁，值得学习。

作为普及组的试题，出题者的意图可能仅仅希望考察选手对搜索和递推算法的掌握，但是，本题作为组合数学 Catalan 数的经典模型，可以用组合数学的方法快捷高效的求解。类似的利用 Catalan 数求解的问题有 NOI2001 福建组队赛《球迷购票问题》等，类似的递推问题还有 NOI2000 福建组队赛《车皮排序问题》等，希望有能力的同学继续研究。

摘要：

	算法一	算法二	算法三
算法	递推	递推	Catalan 数
时间复杂度	$O(n^2)$	$O(n^2)$	$O(n)$
空间复杂度	$O(n)$	$O(n^2)$	$O(1)$

问题转述：

求一列共 n 辆的火车按顺序通过一个栈所产生的排列总数。

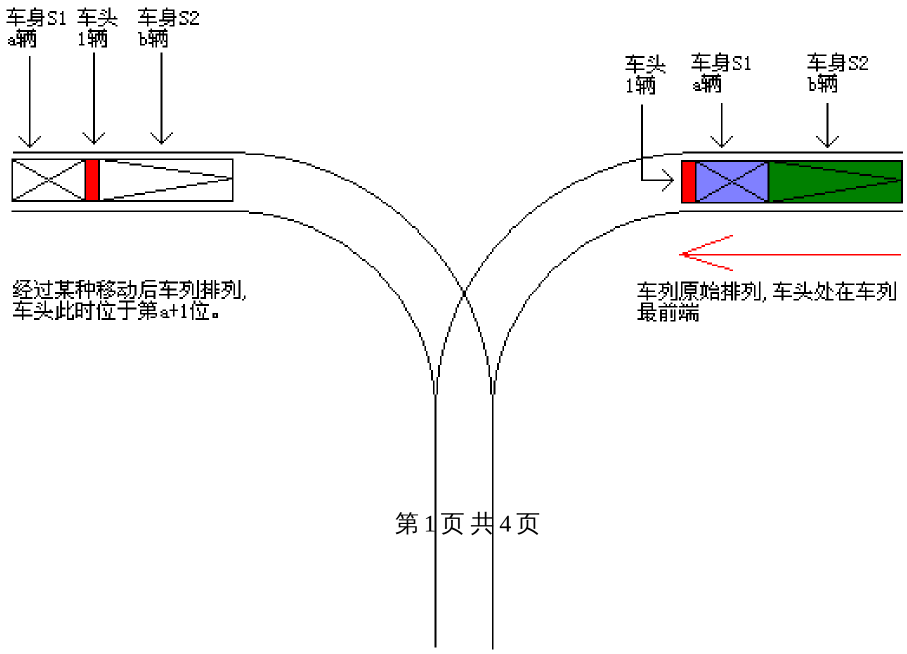
分析：

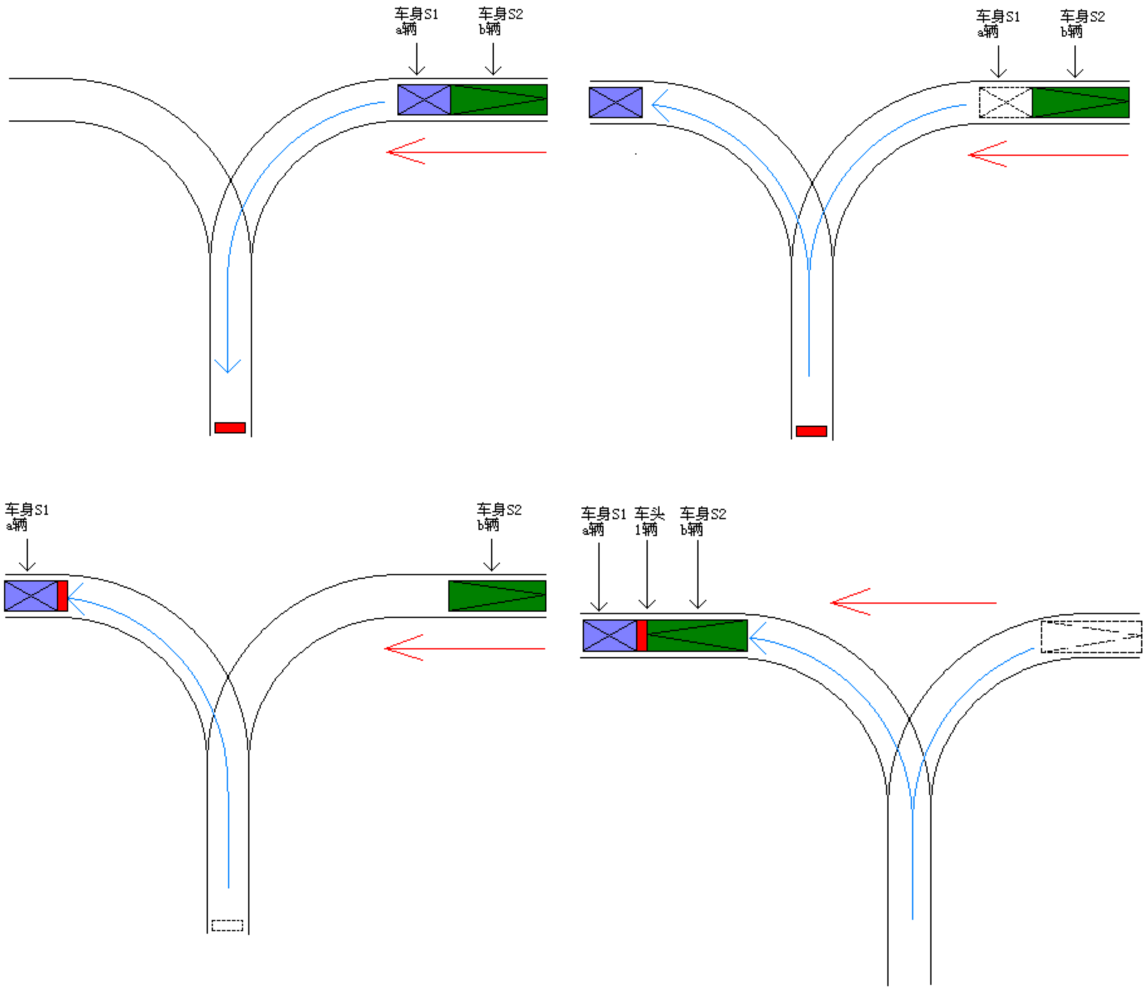
这一类组合计数题目显然不能用搜索的方法把所有可能的移动方案都穷举出来再统计总数——这样做时间复杂度极大。这道题与经典的 HANOI 问题很相似，所以应当根据问题本身的性质，利用组合数学的原理，将原问题转化为递归形式，找到计算总数的递归方程，再进行计算。

算法一：

我们不妨直接设 n 辆火车产生的排列总数为 f(n)。看看能不能找到一些规律。

如图,n 列火车通过栈,起始车头在车列最前端。经过移动后,车头处在了第 a+1 位,车头前有 a 辆车,车头后有 b 辆车(a>=0,b>=0)。则 n=a+b+1,b=n-a-1。





若要达到上述移动目的,步骤为:

(1) 将车头进栈;

(2) 将车头后 a 辆车依次通过栈，移至轨道另一端；

(3) 将车头出栈，则车头恰好排在第 a+1 位；

(4) 将轨道右端剩余 b 辆车依次通过栈，移至轨道另一端；

不难证明，移动方案仅此一种。问题是每个步骤又有许多种不同的移动方法。显然步骤（1）（3）各只有一种移动方法。仔细观察步骤（2）（4）。我们前面定义了“ n 辆火车依次通过栈产生的排列总数为 f(n)”，而步骤（2）恰恰是这个问题的子问题。即步骤二可写为“将 a 辆火车依次通过栈”，根据前面定义，其移动方案总数为 f(a)。同理，步骤(4)的移动方法总数为 f(b)。

根据**乘法原理**，要完成上述工作：

总的方法数 tot=步骤(1)的方法数*步骤(2)的方法数*步骤(3)的方法数*步骤(4)的方法数

=1* f(a) * 1* f(b)

=f(a) * f(b)

=f(a) * f(n-a-1) (因为 b=n-a-1)

我们目前已求得将 n 列火车通过栈，且将位于原车列首位的车头经过移动后位于移动后的车列第 a+1 位的方法总数, 即 **f(a)*f(n-a-1)**。但是原火车头经过移动后可能处在移动后车列的任意一个位置，即 a 的取值是任意的。由于共有 n 辆车，因此移动后原火车头前面

的车数可能有 0~n-1 辆，即 0≤a≤n-1。

要完成某个特定的移动方法，a 只能取某个特定的值。根据**加法原理**，将 n 辆火车依次通过栈的移动总数为：

总的方法数 f(n) = 取 a=0 的方法数 + 取 a=1 的方法数 + ... + 取 a=n-1 的方法数

= f(0)*f(n-0-1) + f(1)*f(n-1-1) + f(2)*f(n-2-1) + … + f(n-1)*f(n-(n-1)-1)

即 $f(n)=\sum_{0\le i\le n-1} f(i)*f(n-i-1)\quad (n\ge 1)$

边界值：**f(0)=1**;

有了以上递归公式，不难用递推的方法写出程序。

算法二：

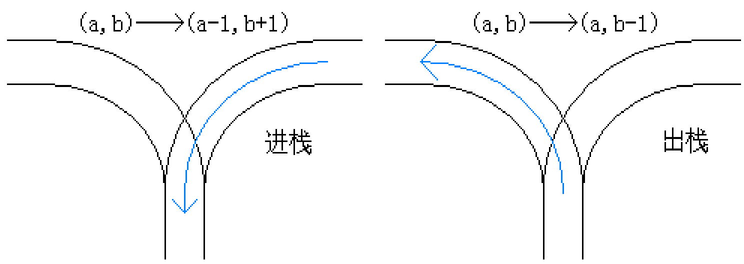
前面所说的搜索法虽行不通，但它也许能给我们一些提示。如果用深度优先搜索(DFS)，穷举所有可能的移动方法来做的话，当搜索到某个状态下，所能做的移动方法无非有两种：（1）将轨道右方的第一列火车进栈；（2）将栈顶的火车出栈，进入左边的轨道。

设此时轨道右方，栈，轨道左方的火车数分别为 **a,b,c**。我们就能用**(a,b,c)**表示出当前的状态。显然 n=a+b+c,则 c=n-a-b。即已知 a 和 b，c 就被确定，所以我们可以用**(a,b)**来作为状态的表示方法。则起始状态为**(n,0)**，目标状态为**(0,0)**。又由上面的两种移动方法。我们可类似的得到两种状态转移方式：

进栈 **(a-1,b+1)** (a>0)

(a,b)

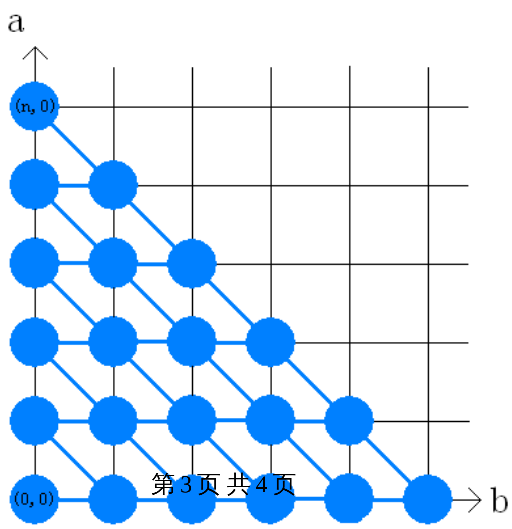
出栈 **(a,b-1)** (b>0)



再设 **f(a,b)**为从状态**(a,b)**通过移动火车变为状态**(0,0)**的所有移动方法。类似于动态规划的状态转移方程，我们可写出以下递归式：

$$\underset{(a+b\le n)}{f(a,b)}=\begin{cases} f(a-1,b+1)+f(a,b-1) & (a>0,b>0) \\ f(a-1,b+1) & (a>0,b=0)\ \text{(此时只能作进栈操作)} \\ f(a,b-1) & (a=0)\ \text{(此时只能作出栈操作)} \end{cases}$$

边界值：**f(0,0)=1**。



按 a 的值从 0~n 划分阶段，亦可通过递推求得 **f(n,0)**的值，即为所求。如果只保存两个阶段进行递推，还可将空间复杂度降为 *O(n)*。这个算法虽然不如算法一简洁，但对于本题来说已经很不错了。而且它在思维难度上要比算法一容易一些。

算法三：

对于算法一中的数列 f(i)来说，每一项都是被唯一确定的。那么我们是否可以找到其通项公式，从而直接算出 f(i)呢？

答案是肯定的。本题实际是一个经典的 Catalan 数模型。有关 Catalan 数的详细解释请参考《组合数学》等书。对于数列 f(i)有公式可直接算出：

$$f(n)=\frac{1}{n}\binom{n}{n+n}=\frac{1}{n}\times\frac{n\times(n+1)\times(n+2)\times\cdots\times(2n-1)\times 2n}{1\times2\times3\times\cdots\times n\times(n+1)}$$

具体实现时，若直接用上述公式计算，对数字的精度要求较高。可将其化为递归式 $f(n)=\frac{2*(2n-1)}{n+1}f(n-1)$ 再进行递推计算，并且注意类型的定义要用 comp 型。

其实，许多看似不相关的问题都和 Catalan 数有密切关系。例如所有节点数为 n 的二叉树的个数就恰为上式中的 f(n)。因此，Catalan 数的应用范围很广。

总结：

做这一类组合数学的问题，需要掌握一些基本的组合数学方面的知识，也要善于对问题作转化处理。组合数学是信息学的基础，也是每一个参赛选手的必修课。在这一题的解题过程中，我们就可以看到组合数学是如何应用的。

参考程序(仅供测试使用)：

算法一	算法二	算法三
1.pas	2.pas	3.pas

测试数据：

1	1
3	5
9	4862
15	9694845
18	477638700