递推法

山东省实验中学　宁华

一、引言

1、人类的递推

递推思想，在实际生活中我们经常用到。人脑在遇到“未知的问题”时，大多数人第一直觉都会从积累的“先验知识”出发，试图从“已知”推导“未知”，最终“解决问题”。

2、计算机的递推

计算机在运用递推思想时，大多都是“重复性”的推理。比方说，从“今天是星期一”推出“明天是星期二”，继而推出“后天是星期三”，而不是从“今天是星期一”一下就推出“后天是星期三”。计算机的推理是一步一步的，每步推理的结构都是相同（十分类似）的，往往可以通过“继而往复”的推理，即把“之前推导的结果”作为“后续推导的开始”，如此下去就可以得到最终的解。

3、递推的含义

所谓递推，是指从初始条件出发，利用某种特定关系，逐步推出中间过程，直至推出最终结果。

（1）其中初始条件或是问题本身已经给定，或是通过对问题的分析与化简后确定。无论怎样，我们如果想让计算机帮助我们解决问题，就需要把这些条件告诉计算机。——递推初值

（2）我们再把“特定关系”——递推关系 告诉计算机，计算机就可以发挥他“善于快速重复工作”的优势，从“初始”开始，逐步推出“结果”并告诉我们。

二、典型题目分析——例 走楼梯问题

1、题目描述：

一个含有 20 阶的楼梯，一次可以走 1 阶或 2 阶，从底走到顶一共有多少种走法？

2、问题分析：

遇到一个问题，如果没有思路，不妨从小规模数据入手，尝试寻找规律。

列表如下：

台阶数	方案	方案数
1	1	1
2	1+1 2	2
3	1+1+1 2+1 1+2	3
4	1+1+1+1 2+1+1 1+2+1 1+1+2 2+2	5
5	1+1+1+1+1 2+1+1+1 1+2+1+1 1+1+2+1 2+2+1 1+1+1+2 2+1+2 1+2+2	8

“数感”比较好的同学，可能已经发现了规律：相邻两项的和恰好等于下一项，这就是著名的斐波那契数列：1　2　3　5　8　13　……

设 fn 表示 n 个台阶的走法总数，则我们可以写出以下式子（递推式）：

$$f_n=\begin{cases}1(n=1)\\2(n=2)\\f_{n-1}+f_{n-2}(n\geq 3)\end{cases}$$

3、递推的两个要素：

（1）递推初值（边界值）

（2）递推式

4、递推式的正确性

当有 1 个台阶的时候，只有 1 种走法，即 f(1)=1

当有 2 个台阶的时候，有 2 种走法，即 f(2)=2

这种小规模数据的答案是我们经过试验得到的，正确性毋庸置疑。

但是当 n 比较大时， f(n) = f(n-1) + f(n-2) 是我们根据小规模数据观察得到的式子，其正确性是否存疑呢？

设 f(n) 表示走到第 n 个台阶的走法总数，则我们可以根据最后一步的走法把走到第 n 个台阶的走法分成两类：

（1）最后一步走了 1 个台阶——上一步在第 n-1 个台阶处

（2）最后一步走了 2 个台阶——上一步在第 n-2 个台阶处

因此，走到第 n 个台阶的走法总数，就等于走到第 n-1 个台阶的走法总数+走到第 n-2 个台阶的走法总数，即：

f(n) = f(n-1) + f(n-2)

从列表上来看，就是这样子的：

台阶数	方案	方案数
1	1	1
2	1+1 2	2
3	1+1+1 2+1 1+2	3
4	1+1+1+1 2+1+1 1+2+1 1+1+2 2+2	5
5	1+1+1+1+1 2+1+1+1 1+2+1+1 1+1+2+1 2+2+1 1+1+1+2 2+1+2 1+2+2	8

当然，也可以根据第一步的走法进行分类。此时，f(n)的含义可以这样来表述：还剩下 n 个台阶的走法总数。

（1）第一步走了 1 个台阶——还剩下 n-1 个台阶

（2）第一步走了 2 个台阶——还剩下 n-2 个台阶

从列表上来看，就是这样子的：

台阶数	方案	方案数
1	1	1
2	1+1 2	2
3	1+1+1 2+1 1+2	3
4	1+1+1+1 2+1+1 1+2+1 1+1+2 2+2	5
5	1+1+1+1+1 1+2+1+1 1+1+2+1 1+1+1+2 1+2+2 2+1+1+1 2+2+1 2+1+2	8

我们可以得到同样的递推式：

f(n) = f(n-1) + f(n-2)

5、代码落实

有了递推式，代码一般就比较容易了。接下来，大家自己写代码并提交吧。

https://www.luogu.com.cn/problem/P1255

注：一定要注重代码落实！通过写代码，你可能会发现新的问题。

三、小结

1、递推：找到“前因后果”，从而“步步为营”。

2、寻找递推式常用方法：

方法 1、观察法：小规模数据找规律，发现解决问题的递推式；

例如：1 3 6 10 15 21 28 ……

f(1) = 1

f(n) = f(n-1) + n （当 n>1 时）

平时注意观察数据，发掘数字之间的关系，提高自己的“数感”。

方法 2：分解问题（分类、分步）：通过分解问题，发掘前后之间的联系，

通常以“第一步”或“最后一步”进行分类或分步，采用加法原理或乘法原理建立前后项之间的联系，进而写出递推式。

3、基本概念：

状态——对所面临问题的规模、性质等的描述.

f(n)——数学中的“函数”

n——数学中的“自变量”

根据问题，定义状态的含义：

比如，走楼梯问题中，可以设 f(n)表示：

(1) 走到第 n 个台阶的走法总数

(2) 前 n 个台阶的走法总数

(3) 还剩下 n 个台阶的走法总数

对于本题而言，这三种定义本质上是相同的。但有些题目，含义不同，其对应的值也就不同。另外，状态含义定义好之后，在后续使用时要紧扣其定义，如果发生状态转移困难，可以调整其含义。

4、递推解题的几个关键点

（1）明确状态

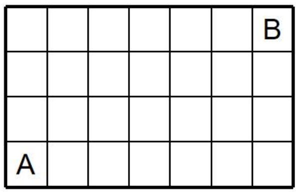
①确定参数及状态的含义，确立边界值及目标状态。

根据状态的复杂程度，确定合适的参数来描述。

例如：

走楼梯问题：f(n)表示 n 个台阶的走法总数，只需要 1 个参数就可以。

走网格问题：n 行 m 列的网格，只能向上或向右，从左下角走到右上角的走法总数。



任务：独立思考，尝试写出递推式。

分析：走网格相当于一个二维的走楼梯问题，此时定义状态就需要两个参数了。另外，状态的含义也可以有多重定义，比如：

f(x,y)表示：从起点走到(x,y)的方案数，则边界值为 f(1,1)=1，目标状态为：f(n,m)

若 f(x,y)表示：从(x,y)走到终点的方案数，则边界值为 f(n,m)=1，目标状态为：f(1,1).

②有些问题可能需要辅助状态

例：在所有的 5 位数中，有多少个数中有偶数个数字 3 ？例如，10000，23333，30003 都是满足条件的数。

任务：独立思考，尝试写出递推式。

分析：可以设 f(n)表示 n 位数中有偶数个 3 的数的个数，这样我们的目标是 f(5)。

5 位数中有偶数个 3 的数，可以这样得到：

（Ⅰ）最高位不是 3, 剩余 4 位中有偶数个 3;

（Ⅱ）最高位是 3, 剩余 4 位中有奇数个 3.—那么此时 4 位数中有奇数个 3 的数的个数这个“因”就需要被我们保存下来, 所以可以增设一个状态 g(n)表示 n 位数中有奇数个 3 的数的个数. 而我们的目标仍是 f(5). 或者将状态 f(n)增加一维, f(n,0)表示 n 位数中有偶数个 3 的数的个数, f(n,1)表示 n 位数中有奇数个 3 的数的个数. 这样我们的目标是 f(5,0).

③有些状态包含隐藏信息

例如上面求 5 位数中有偶数个 3 的数的个数，最高位不能为 0 就是当状态为 5 位数时的隐藏信息，而在 4 位数、3 位数的状态描述里，最高位则可以为 0。

（2）边界条件

边界条件，是递推进行的前提，边界值必须保证正确，否则就会一错到底。

边界条件，有的具有实际的意义，比如，爬 1 个台阶有 f(1)=1 种方法，爬

2 个台阶有 f(2)=2 种方法，利用这两个边界值可以推出后续的结果。但也有的边界条件并不具备实际意义，需要自己通过分析去设定边界值。

例：放苹果问题：把 M 个同样的苹果放在 N 个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的放法？

任务：独立思考，状态如何描述，据此分析边界值是什么？

分析：这里有苹果、盘子，所以状态设定需要两个参数。可以设 f(m,n)表示把 m 个同样的苹果放在 n 个同样的盘子里的不同放法总数。则边界条件是 f(x,0)、f(0,y)，这些边界值如何设定呢？比如 f(x,0)，把 x 个苹果放在 0 个盘子，没有盘子，从现实意义上来看，f(x,0)的值应为 0，但作为递推初值的话，其值则需要设置为 1.这个时候，边界值的设定，可能已经失去了现实意义，仅仅是为了完成后续递推所设定的特定值。

（3）递推式

也就是确定当前要求的 f(i) 依赖于之前的哪些 f(j)

此时，我们心中必须明确一点：求解“后果”的前提是：“前因”都已找到，此时我们只关注“前后之间的因果关系”，而不关注前因是如何得到的。

比如，此刻，我们要求 f(10)，那么，f(0)，f(1)，f(2)，...，f(9)的值都已经有了，我们现在要做的是寻求 f(10)和他们之间的关系，进而去求解 f(10)

写出递推式，就是要找出当前状态 i 下的 f(i) 与之前状态 j 下的 f(j) 之间的关系，我们是在假设所有小于 i 的 j 的 f(j)都已经具有了确定的值，进而去推出 f(i)，所以这个时候我们关注的是 f(i) 与 f(j) 之间的关系，而不会考虑 f(j) 的值是怎么求出来的。

5、学习递推，要注重思考

对于递推问题，平时学习要注重思考，多角度考虑解决问题的方法，培养发散思维能力。

比如：走楼梯问题，我们以“第一步”或“最后一步”所走台阶数不同为依据进行分类，找到了当前项与之前项之间的关系。你还可以提出其他的分类依据吗？

比如，我提出了以下分类依据：以某个特定的台阶是否经过作为分类依据。

假如现在要计算 10 个台阶的走法。我们指定第 3 个台阶为特定台阶，则走到第 10 个台阶，要么经过第 3 个台阶，要么不经过第 3 个台阶。（没问题吧）

设 f(n)表示：共 n 个台阶的走法总数，则：

（1）经过第 3 个台阶：

此时整个过程分成了两大步：

①从底走到第 3 个台阶，共 3 个台阶，有 f(3)种走法。

②从第 3 个台阶走到第 10 个台阶，共 7 个台阶，有 f(7)种走法。

根据乘法原理，有：f’(10) = f(3)*f(7)

（2）不经过第 3 个台阶：

此时整个过程也分成了两大步：

①从底走到第 2 个台阶，共 2 个台阶，有 f(2)种走法。

②从第 2 个台阶跨越两个台阶走到第 4 个台阶，再从第 4 个台阶走到第 10 个台阶。这个过程其实也分成两步，从第 2 个台阶到第 4 个台阶只有 1 种走法，然后从第 4 个台阶到第 10 个台阶，共 6 个台阶，有 f(6)种走法。

所以有：f”(10)=f(2)*(1*f(6))=f(2)*f(6)

综上，f(10)= f(3)*f(7) + f(2)*f(6)

而实际上，指定的台阶可以是第 1 ~ 9 之间的任意一个台阶。如果我们指定台阶为第 1 个台阶，我们发现，经过第 1 个台阶和不经过第 1 个台阶，恰好就对应我们刚才所说的第一步走 1 个台阶和走 2 个台阶的情况。

推广到一般情况，递推式可以写作：

f(n) = f(t)*f(n-t) + f(t-1)*f(n-t-1) 其中 t 的值满足 1<=t<=n-1，边界条件 f(0)=f(1)=1

至于 t ，只要在求 f(i)的时候指定一个 1<=t<=i-1 的值即可，只要在此范围内，每次 t 都随机取也可。

大家不妨编写代码测试一下吧~

代码仅供参考:

```
#include<bits/stdc++.h>

using namespace std;

int f[25];

int main()

{

   srand(time(0));

   f[0]=1;

   f[1]=1;

   for(int i=2;i<=20;i++)

   {

       int t=rand()%(i-1)+1;

       f[i]=f[t]*f[i-t]+f[t-1]*f[i-t-1];

       cout<<f[i]<<" ";
```

```
    }
    return 0;
}
```

我们讲的这种“指定特定元素”分解问题的方法是否具有推广意义呢？不妨假设每步的走法除了可以一步走 1 个台阶和 2 个台阶，还可以一步走 3 个台阶，此时仍然求 10 个台阶的走法总数，请大家利用这种方法，自己写一下递推式，并编程进行测试。相信大家会有非常的感受与收获。

四、继续走楼梯

你还可以提出其他的方法吗？

1、一种新的方法

例如，我又提出了这样一种分类方法：

“第一次一步跨越两个台阶”行为出现在第几步

则可能有：

“第一次一步跨越两个台阶”行为出现在第 1 步；

“第一次一步跨越两个台阶”行为出现在第 2 步；

“第一次一步跨越两个台阶”行为出现在第 3 步；

……

“第一次一步跨越两个台阶”行为出现在最后一步(第 n-1 步)；

请思考以上分类是否正确？

分析：

遗漏了一种情况：“第一次一步跨越两个台阶”行为没有出现

在“第一次一步跨越两个台阶”行为出现之前，肯定都是一步一个台阶地走上来的，也就是之前的走法只有 1 种，我们只需要考虑剩下台阶的走法即可。

由此：

“第一次一步跨越两个台阶”行为出现在第 1 步；——还剩下 n-2 个台阶

“第一次一步跨越两个台阶”行为出现在第 2 步；——还剩下 n-3 个台阶

“第一次一步跨越两个台阶”行为出现在第 3 步；——还剩下 n-4 个台阶

……

“第一次一步跨越两个台阶”行为出现在最后一步(第 n-1 步)；——还剩下 0 个台阶

“第一次一步跨越两个台阶”行为没有出现——全都是一阶一阶走上来的

此处定义 f(n)的含义是：n 个台阶的走法总数。那么对于特定的 n，f(n-1)就是非法状态，因为对于 n 个台阶，不可能出现了“一步跨越两个台阶的行为”后还剩下 n-1 个台阶的情况。

任务：独立思考，写出递推式。

分析：据此写出递推式：f(n) = f(n-2) + f(n-3) + …… + f(0) + 1

边界条件 f(0)=1, f(1)=1

注意：

虽然对特定的 n 来说，f(n-1)是非法状态，但对后续的 n，它又变成了合法状态。比如，对于 f(5)，推它的值时用不到 f(4)，但在后续推 f(6), f(7), ……时会用到，所以我们递推过程中对这些状态的值都需要求出来，以备后用。

上述递推式也可修改为：

f(n) = f(n-2) + f(n-3) + …… + f(0)

边界条件 f(0)=2, f(1)=1

这里的 f(0)=2 指当有 0 个台阶时有 2 种走法。这个边界值可能不具备现实意义，只是为了我们后续推导设定的特定值。实际上，如果非要讲道理，也可以这样解释：0 个台阶，我以一步走 1 个台阶的方式走了 0 步，或者一步走 2 个台阶的方式走了 0 步，所以有 2 种走法。这样“强词夺理”也勉强可以让人接受。

这个递推式是否正确呢？

可以编程进行测试~

代码仅供参考：

```
#include<bits/stdc++.h>
using namespace std;
int f[25];
int main()
{
   f[0]=2;
   f[1]=1;
   for(int i=2;i<=20;i++)
   {
      for(int j=0;j<=i-2;j++)
         f[i]+=f[j];
      cout<<f[i]<<" ";
```

```
    }
    return 0;
}
```

2、类比得新式

刚才以“第一次一步跨越两个台阶”行为出现在第几步为分类依据，那当然也可以以“第一次一步跨越 1 个台阶”行为出现在第几步作为分类依据。

任务：独立思考，写出递推式。

分析：以“第一次一步跨越 1 个台阶”行为出现在第几步作为分类依据。

则可能有：

“第一次一步跨越 1 个台阶”行为出现在第 1 步；——还剩下 n-1 个台阶

“第一次一步跨越 1 个台阶”行为出现在第 2 步；——还剩下 n-3 个台阶

“第一次一步跨越 1 个台阶”行为出现在第 3 步；——还剩下 n-5 个台阶

……

“第一次一步跨越 1 个台阶”行为出现在最后一步(第 x 步)——还剩下 0 个台阶，此时会引出一个新问题：当 n 是奇数时，这种情况是存在的，但 n 是偶数时这种情况不存在。

“第一次一步跨越 1 个台阶”行为没有出现——全都是两阶两阶走上来的，自然引出一个新问题：n 是偶数时，这种情况是存在的，但 n 是奇数时，这种情况不存在。

由此写出递推式：

当 n 是奇数时：

$f(n) = f(n-1) + f(n-3) + f(n-5) + \cdots + f(2) + f(0)$

当 n 是偶数时：

$f(n) = f(n-1) + f(n-3) + f(n-5) + \cdots + f(1) + 1$ （最后加上的 1 就是“一步跨越 1 个台阶”行为没有出现”的情形）

边界值：$f(0)=f(1)=1$

请同学们编写程序进行验证，看得到的答案与刚才是否一致。

3、小结

（1）同样的问题可能有多个看上去不同的递推式

同样是斐波那契数列：

$f(n) = f(n-1) + f(n-2)$ 边界条件 $f(1)=1$，$f(2)=2$

$f(n) = f(t)*f(n-t) + f(t-1)*f(n-t-1)$ 其中 t 的值满足 $1<=t<=n-1$，边

界条件 f(0)=f(1)=1

f(n) = f(n-2) + f(n-3) + …… + f(0) + 1 边界条件 f(0)=1, f(1)=1

（2）有些边界值的设定可以根据实际问题自行调整

f(n) = f(n-2) + f(n-3) + …… + f(0) 边界条件 f(0)=2, f(1)=1

（3）问题分解（分类、分步）时要不错、不重、不漏

比如，以“第一次一步跨越两个台阶”行为出现在第几步作为分类依据时，

“第一次一步跨越两个台阶”行为没有出现这种情况就不能忽略。

（4）分类/分步标准不同，会导致解决问题的复杂度不同

（5）平时学习要注重思考，比如，走楼梯问题，你还能找到其他的递推式吗？

五、一些经典的递推问题

（一）、斐波那契数列

（二）、汉诺塔问题

（三）、分割问题

（四）、卡特兰数

（五）、第二类斯特灵数

（六）、错排问题

（七）、一类博弈问题

……

以下内容，请结合刚才介绍的方法，自主探究学习。

（一）、斐波那契数列

斐波那契数列（Fibonacci sequence），又称黄金分割数列，因数学家莱昂纳多·斐波那契（Leonardoda Fibonacci）以兔子繁殖为例子而引入，故又称为“兔子数列”。原问题是这样的：

有人想知道一年内一对兔子可繁殖成多少对，便筑了一道围墙把一对兔子关在里面。已知一对兔子从第三个月开始，每一个月可以生一对小兔子，而一对兔子出生后，第三个月开始又可以生一对小兔子。假如一年内没有发生死亡，则一对兔子一年内能繁殖成多少对？

请大家利用刚才介绍的方法，自己写出递推式，并解释其正确性。

【课后练习】

1、骨牌覆盖问题：在 2×n 的长方形方格中，用 n 个 1×2 的骨牌铺满方格，输入 n ，输出铺放方案的总数。

2、一枚均匀的硬币抛掷 n 次，从不接连出现正面朝上的情况一共有多少种？

比如，n=3 时，有 5 种合法的情况：

反反反

反反正

反正反

正反反

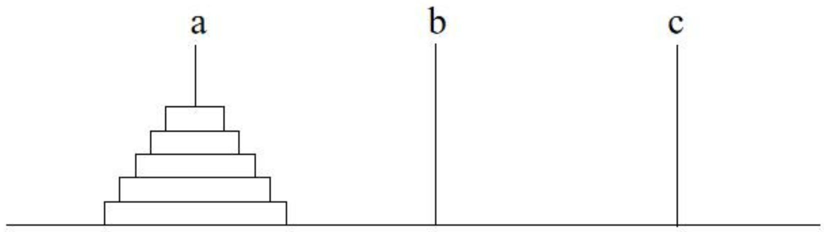
正反正

3、自然数从 1 到 N 按照顺序排成一排，可以从中取走任意数，但是相邻的两个数不可以同时被取走。一共有多少种取法？注：不取也算一种取法。

例如，N=3 时，有 5 种取法：不取, 取 1, 取 2, 取 3, 取 1 和 3。

(二)、汉诺塔问题

Hanoi 塔由 n 个大小不同的圆盘和三根木柱 a,b,c 组成。开始时，这 n 个圆盘由大到小依次套在 a 柱上，如图所示。



要求把 a 柱上 n 个圆盘按下述规则移到 c 柱上：

(1)一次只能移一个圆盘；

(2)圆盘只能在三个柱上存放；

(3)在移动过程中，不允许大盘压小盘。

问将这 n 个盘子从 a 柱移动到 c 柱上，最少需要移动多少次盘子？

分析

1、观察法

小规模数据找规律

2、分治法

递推式： $f(n) = 2f(n-1) + 1$ 边界条件 $f(1)=1$

3、递推式与通项式

n 个盘子的 hanoi 问题，最少移动多少次？

通项公式：$f(n)=2^n - 1$

证明：

数学归纳法

$f(1)=1$ 成立；

假设 $f(n)=2^n - 1$

$f(n+1)$

$=2f(n)+1$

$=2(2^n-1)+1$

$=2^{(n+1)} - 1$

证毕。

方法 2：由递推式得通项公式

$f(n)=2f(n-1)+1$

$f(n)+1=2(\ f(n-1)+1\ )$

$f(n)+1=(f(1)+1)*2^{(n-1)}=2^n$

$f(n)=2^n - 1$

说明：

我们用递推法求解问题时，关键是找出后项与前项之间的数学关系。然后从初始项开始，逐项递推。在算出前 n-1 项之前，我们是不知道第 n 项的值的。可以说，为了求第 n 项值，我们总共要计算 n 次，因此时间复杂度为 O(n)

而通项式建立的是第 n 项与项数 n 之间的数学关系。我们可以直接用项数 n 计算出第 n 项的值。可以说，为了求第 n 项的值，我们只需计算 1 次，因此时间复杂度为 O(1)(不绝对).

【课后练习】

Hanoi 塔由 n 个大小不同的圆盘和三根木柱 a,b,c 组成。开始时，这 n 个圆盘由大到小依次套在 a 柱上，如图所示。

要求把 a 柱上 n 个圆盘按下述规则移到 c 柱上：

(1)一次只能移一个圆盘；

<u>(2)圆盘只能在相邻柱子间移动；</u>

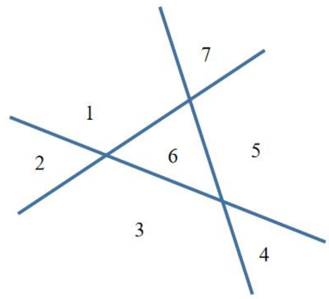
(3)在移动过程中，不允许大盘压小盘。

问将这 n 个盘子从 a 柱移动到 c 柱上，最少需要移动多少次盘子？

(三)、分割问题

例：直线分割

n 条直线，最多能把平面分割成多少个部分？



样例输入：

3

样例输出：

7

分析

递推式

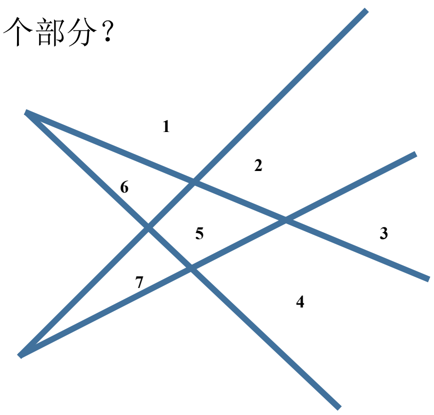
f[n]=f[n-1]+n (n>0)

f[0]=1

【课后练习】找出以下每个问题的递推式和通项式，并编程实现。

1、折线分割

n 条折线，最多能把平面分割成多少个部分？



样例输入：

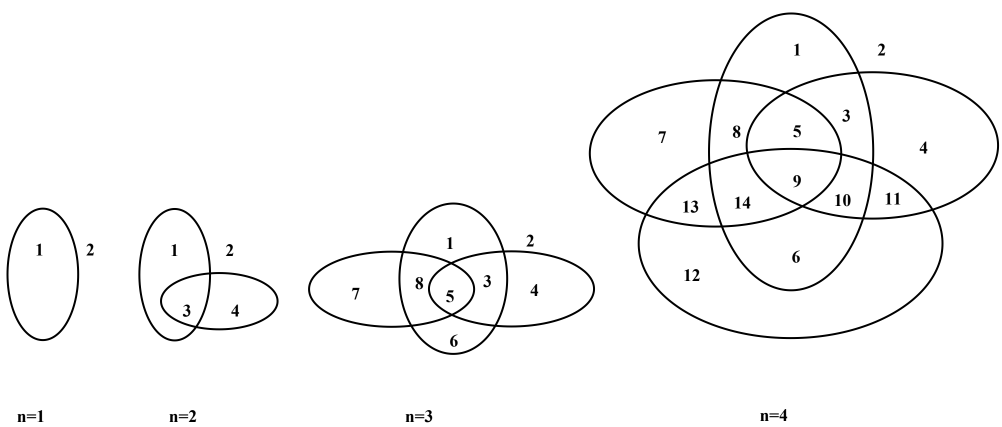
2

样例输出：

7

2、曲线分割

设有 n 条封闭曲线画在平面上，而任何两条封闭曲线恰好相交于两点，且任何三条封闭曲线不相交于同一点，求这些封闭曲线把平面分割成的区域个数 m。



样例输入：

2

样例输出：

4

3、区域划分问题

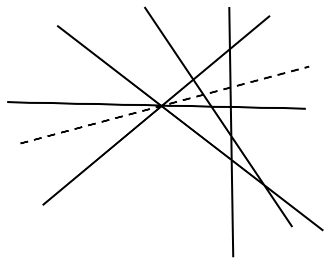
同一平面内的n(n<=500)条直线,已知其中有p(p>=2)条直线相交于同一点，则这n条直线最多能将平面分割成多少个不同的区域?

输入：

n p (输入保证n>=p)

输出：

区域个数



样例输入：

3 2

样例输出：

7

4、立体切割问题

例：n刀最多可以把一个西瓜切成几块？

样例输入：

3

样例输出：

8

(四)、卡特兰数

例 NOIP2015提高组初赛的一道题目

结点数为 5 的不同形态的二叉树一共有______种。

分析:

h(0)=1, h(1)=1

h(n)= h(0)*h(n-1) + h(1)*h(n-2) + ... + h(n-1)h(0) (n>=2)

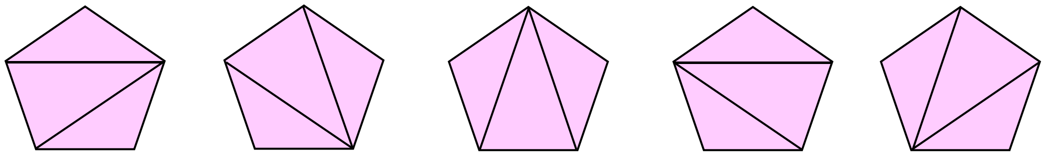
【课后练习】

1、凸多边形的三角形剖分问题

在一个凸 n 边形中，通过不相交于 n 边形内部的对角线，把 n 边形剖分成若干个三角形。

输入 n，问有多少种不同的剖分方案？

如 n=5 时有 5 种剖分方案:



2、买票问题

2n 个人排队买票,其中 n 个人持有 100 元钞票，而其他 n 个人持有 50 元钞票。假如售票处没有预备零钱,而每张票卖 50 元。问有多少种排队次序使得不会出现无法找零的问题?

3、括号匹配问题

n 对括号有多少种匹配方式？

例如：

0 对括号：[空序列] 1 种

1 对括号：() 1 种

2 对括号：()()、(()) 2 种

3 对括号：((()))、()(())、()()()、(())()、(()()) 5 种

那么问题来了，n 对括号有多少种正确配对的方式呢？

4、在一个 n*n 的地图中，只能在下三角行走，不能越过对角线，每次向右或向上走一格，从左下角到右上角有多少种走法？

小结：Catalan 数

上面一些问题有些是同构的，但有些却实在看不出联系来，他们的答案却都为卡特兰数。在《Enumerative Combinatorics》一书中，竟然提到了多达 66 种组合问题和卡特兰数有关。

注：Catalan 的递推式与通项式

令 h(0)=1, h(1)=1，Catalan 数满足递推式

h(n)= h(0)*h(n-1) + h(1)*h(n-2) + ... + h(n-1)h(0) (n>=2)

Catalan 通项公式为

h(n)=C(2n,n)-C(2n,n-1)　　(n=0,1,2,...)

h(n)=C(2n,n)/(n+1)

另类递推式

h(0)=1

h(n)=h(n-1)*(4n-2)/(n+1)　　(n=1,2,...)

(五)、第二类 stirling 数

n 个有区别的球放到 m 个相同的盒子中，要求无一空盒，其不同的方案数用 S(n,m)表示，称为第二类 Stirling 数。

输入 n、m，输出 S(n,m)

分析

设有 n 个不同的球，分别用 b1,b2,……bn 表示。从中取出一个球 bn，bn 的放法有以下两种：

①bn 独自占一个盒子；那么剩下的球只能放在 m-1 个盒子中，方案数为 S(n-1,m-1)；

②bn 与别的球共占一个盒子；那么可以事先将 b1,b2,……bn-1 这 n-1 个球放入 m 个盒子中，然后再将球 bn 可以放入其中一个盒子中，方案数为 m*S(n-1,m)。

综合以上两种情况，可以得出第二类 Stirling 数定理：

【定理】S(n,m)=m*S(n-1,m)+S(n-1,m-1) (n>1,m1)

边界条件可以由定义 2 推导出：

S(n,0)=0；S(n,1)=1；S(n,n)=1；S(n,k)=0(k>n)。

【课后练习】

放苹果

描述

把 M 个同样的苹果放在 N 个同样的盘子里，允许有的盘子空着不放，问共有多少种不同的分法？（用 K 表示）注意：5，1，1 和 1，5，1 是同一种分法。

输入

二个整数 M 和 N，以空格分开。1<=M，N<=10。

输出

输出相应的 K。

样例输入

7 3

样例输出

8

(六)、错排问题

尼古拉·伯努利给欧拉写了 n 封信件，对应 n 个信封，然而粗心的秘书却把所有信件都装错了信封，那么一共有多少种装错的装法？

分析

递推式：$D_n = (n-1)(D_{n-2}+D_{n-1})$　　特殊地，$D_1=0$，$D_2=1$

注：理解问题的本质

有 n 封信，其中第 1 封信不能装到第 x1 个信封里，第 2 封信不能装到第 x2 个信封里，第 i 封信不能撞到第 xi 个信封里。其中 x1,x2,...,xn 对应 1~n 的一个排列。问有多少种装法？

(七)、一类博弈问题

有一盒火柴共 100 根，甲、乙两人轮流取，每人每次只能取 1 或 2 根，不能不取。如果轮到某人时，其没有火柴可取，则判其输。若甲先取，问其是否有必胜策略？说明原因。

【课后练习】

有一盒火柴共 100 根，甲、乙两人轮流取，每人每次取的火柴数不能超过当前火柴数的一半，不能不取。如果轮到某人时，按规则无法取火柴，则判其输。若甲先取，问其是否有必胜策略？说明原因。

六、总结：抓住本质，灵活应用

递推算法以初始（起点）值为基础，用相同的运算规律，逐次重复运算，直至运算结束。

递推的本质是按规律逐次推出前（后）一步的结果。

递推题目呈现形式千变万化，但万变不离其宗，这个“宗”就是：事物发展前后之间的关系。如果我们能够抓住问题的本质，找出前后之间的关系，无论是顺推还是倒推，写出递推式并不困难。

要想掌握递推方法，还需要做大量的题目，慢慢体会其中的奥妙~

七、上机训练

洛谷：

P1002 [NOIP2002 普及组] 过河卒

P1028 [NOIP2001 普及组] 数的计算

P1044 [NOIP2003 普及组] 栈

P1025 [NOIP2001 提高组] 数的划分

P1096 [NOIP2007 普及组] Hanoi 双塔问题

P1595 信封问题

P1990 覆盖墙壁

一本通：

1312：【例 3.4】昆虫繁殖

1313：【例 3.5】位数问题

1195：判断整除

1196：踩方格