3.琅泽阵(*Langtze Problem*)

通过题目的详细提示，我们知道，A规律对上一层的描述是单一的，B规律对上一层的描述是连续的。

什么意思呢？例如上一层是1113，如果这一层遵循A规律，则对上一层的每一个数字都描述一下，即：11111113，若本层遵循B规律，则对上一层的连续相同数字进行描述，那么将是：3113。

搞清楚了A与B规律，那么我们就得试着去找琅泽阵第*i*层中有多少个*x*的规律了。

我们可以以初始数据为*a*来构建一个琅泽阵。

*a*

1 *a*

1 1 1 *a*

3 1 1 *a*

1 3 1 1 1 1 1 *a*

1 1 1 3 5 1 1 *a*

...

此时，不难证明对于任何*a*>1(*a*≠3,*a*≠5,*a*≠7)都满足每一层仅有1个*a*。那么*a*这个东西影响不大，毕竟说过*a*不会是1了。而且问题中没有涉及到询问1的个数，因此我们可以忽略对1的讨论。下面我们来讨论3。

事实上，对于任何*a*>1都满足琅泽阵**仅存在1个3**（*a*=3时除外）。证明如下：

在第3行后，若出现新的3，则需满足在A规律中出现3个单独连续的1（即这3个1左右两边都没有1），那么在此又往上一次的A规律中必须有两个连续的非1数，然而这并不可能，因为在分的时候，若出现两个连续的*k*和*n*（*k*,*n*>1），则可能为

①*k*个*n*，与该规律矛盾！

②*q*个*k*，*n*个*p*，又与之矛盾！

故往后不可能再出现新的3。

而第四行出现的3是特殊情况，因为前两次的分恰好满足*a*单独存在，导致前一次出现连续三个1，而往后就没有了。

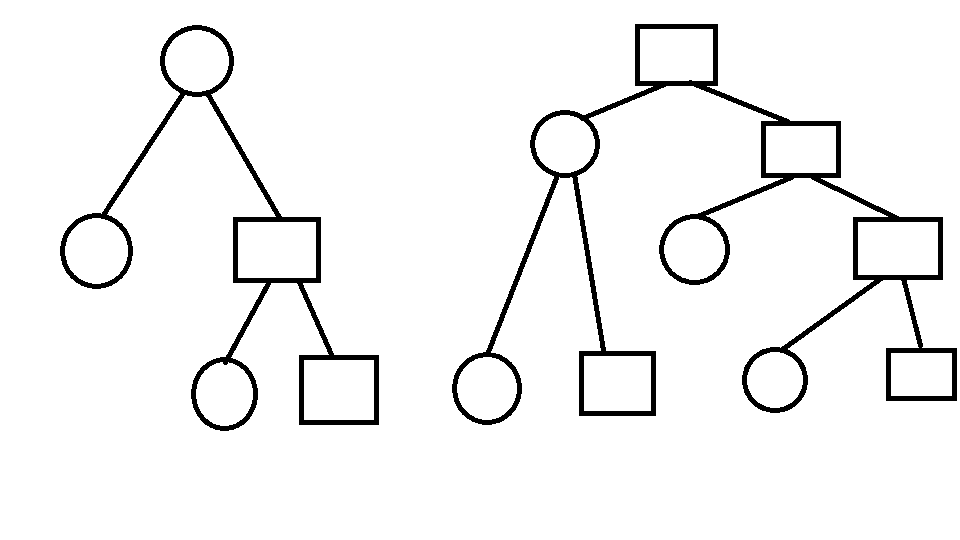
3处理完之后就是5和7了，除此之外，再也没有其它可讨论的了，因为琅泽阵中只会存在1,3,5,7,*a*，这个不难证明。

那么5和7怎么办呢？我们可以往后面继续列，写一下规律，我这里只看B规律。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 潜5 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 |
| 5数量 | 0 | 1 | 2 | 4 | 7 | 12 | 20 | 33 | 54 |
| 7数量 | 0 | 0 | 1 | 2 | 4 | 7 | 12 | 20 | 33 |
| 潜7 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 |

·潜5表示下一次B规律将要增加的5，潜7代表往后第二次B规律将要增加的7.

我们发现了斐波那契数列！！！为什么会是这样呢？

事实上，5是由上一次B规律中两个单独连续的1演变而来的，演变来之后随之又增加了两个单独连续的1，也就是上一次B规律带来了1个5并埋下了一个潜5，那位什么是斐波那契数列呢？我们又发现7是由上两次B规律中两个连续的不等数得来的。而第4行由于3在在前面，可以视为前面有一个与它不等的数，因此埋下了一个潜7，在下一次B规律中，7并没有马上出现，而是演变成了3个1，此时由上一层演变而来的5又与3靠在了一起，埋下一个潜7，那么该行就有一个潜5一个潜7。也就是说，每一个潜5都会带来1个潜5和1个潜7，而不难发现，每一个潜7又会给下两行带来一个潜5和潜7。那么我们知道第四行是1个潜5和1个潜7，第六行由第四行的潜5得到1个潜5和1个潜7，而第八行由第六行的潜5得1个潜5和1个潜7，再由第四行的潜7得到1个潜5和1个潜7...以此类推，也就是说，1个潜5由第2*n*（*n*≥3）行往2(*n*+1)行传递潜5并增加1个潜7，而2(*n*-1)行往2(*n*+1)行传递潜7并增加1个潜5.

如图所示：

可以理解为方框（潜5）是用来【保持】

的，而圆圈（潜7）是用来【提升】的。

那么斐波那契数列也就构成了。

其实这不就是**斐波那契的兔子问题**吗，潜7

是刚出生的，一个月长大，下个月再生兔子。

**总的来说，潜5和潜7在每一层都是相等的，而潜5对后面没有影响，每个潜7对其后面第二次B规律有影响，即F[*n*]=F[*n*-1]+F[*n*-2]。**

我们设2*n*(*n*≥3)层的5的数量为five[*k*]（*k*=*n*），设斐波那契数列为F[*r*]（F[1]=1,F[2]=1），那么2*n*层对应的潜5或潜7就是F[*n*]=F[*k*-1]。则易得：

five[*k*]=five[*k*-1]+F[*k*-2];（将层数除以2是为了便于程序编写，况且A规律中的5和7数量不变）

验证一下，five[5]为第10层的5的数量，five[5]=five[4]+F[3]=2+2=4。

将式子化为斐波那契数列之和得到第*k*层5的数量的**答案式**：



因此我们构建程序就不难了，那么7呢，**7的数量和上一次相同规律时的5的数量是一样的**。

现在讲讲程序解答。

众所周知，F[*n*]=F[*n*-1]+F[*n*-2]，这样的递推式显然不难想到。运用该递推式，大概能得30分的样子。为什么？我们可以看到five[*k*]的答案式展开后运算量及其庞大，特别是当*i*达到一定的数量级的时候，而且程序限时又很紧，所以这样的方法是行不通的。

那么，怎么办呢？

注意，你需要知道斐波那契数列有两个求和公式：





这两个公式分别是斐波那契数列的**奇数项和偶数项求和公式**。

有了它，你还是得不到高分，你还是只有六七十分。。。为什么，因为你求的时候还是会超时，因此你需要一个**二倍项公式**：



可以化成：



运用上述公式，运算量大大减少，**100分*AC***。

详情请见标程：LangtzeAnswer.cpp

题目为原创题，算法为斐波那契数列公式。