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 $a>1(a\neq3,a\neq5,a\neq7)$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

①k 个 n，与该规律矛盾！

②q 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。

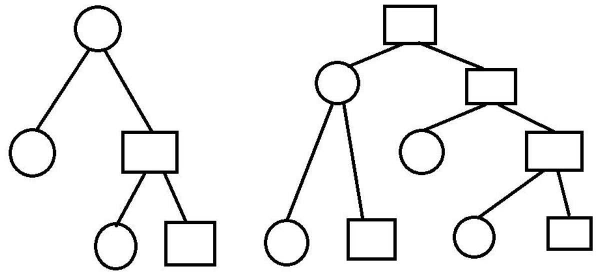
	4	6	8	10	12	14	16	18	20
潜 5	1	1	2	3	5	8	13	21	34
5 数量	0	1	2	4	7	12	20	33	54
7 数量	0	0	1	2	4	7	12	20	33
潜 7	1	1	2	3	5	8	13	21	34

·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 $2n$（$n\geq3$）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 $2n(n\geq3)$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 $2n$ 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**：

$$five[k]=\sum_{i=1}^{k-2}F[i]$$

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：

$$F[2n]=F[2n-1]+\cdots+F[3]+F[1]$$

$$F[2n+1]-1=F[2n]+\cdots+F[4]+F[2]$$

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**：

$$F[2n]=F[n-1]F[n]+F[n+1]F[n]$$

可以化成：

$$F[2n]=F[n]^2+2F[n]F[n-1]$$

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。