快速幂算法，什么是快速幂算法呢？简而言之就是如何快速的计算一个数的幂。

正常的一个个乘的方法是O(N)级别的，我们怎样能够让其更快呢？我们可以将其拆分为O(logN)的，且看怎么做吧。

本来是想拿数据来证明一下确实是快了不少的说，但是一番实验之后确实快了一些，但是和理想中的还是有些差距，所以在这里就不进行数据上的对比了。

大概讲一下原理。我们可以展开为，讲其上的指数转化为二进制的，并且拆开

然后我们就可以这样子计算，将本来是转化为如上的式子，因为展开之后A上面的指数必然为之前的2的幂，所以在写代码计算的时候就将一个A不断平方，然后到bn为1的时候就是有，然后乘上去，其实在这里我也发现上面的式子不是很正确，不过懒得更改了，直接举个例子大致就懂了。

计算

我们可以转化为

然后这个样子就能够进行快速计算了，同时如果遇到需要算大数字的模的时候，可以直接把需要相乘的几个数字在中途模就好了，这样子又不会超出需要的限制。

而这个快速幂算法可以推广到快速幂矩阵算法上面来，具体有什么作用呢？其实我也不是很清楚，但是可以解决计算斐波那系数的大数问题。

我们都知道斐波那系数是0，1，1，2，3，5，8……后面的数字都是前面两个数字的和，如果我们这般设置矩阵X

那么实际上下一个数字就是这两个的和，那我们可以构建一个这样子的矩阵如：

因为word中好像打不出矩阵，就算了，然后在这个就能够做到使得，然后当需要计算的时候：

所以其实就是计算A的n-1次幂就能够计算出Fn出来