在信息安全与数学基础课上学习了不相交循环的知识，当时在课上简单介绍了完美洗牌的知识。所谓的完美洗牌，就是指将一副扑克牌每张牌的顺序完全打乱，使每张牌的位置与之前完全不同，这在许多扑克牌游戏和密码学中得到了广泛的应用。

当时对密码学算法这方面的知识非常感兴趣，我也自己动手编写过欧几里得乘法逆的程序，但这次想自己学着编写一个完美洗牌的程序，算法是这个程序额度核心，完美洗牌的算法有很多，我课后在图书馆查找了很多资料，也从一些算法的论坛上看过源代码，方法有很多，但是不需要依赖于什么特殊的结构，主要是这种算法的思想很方便快捷，我采用的算法核心的公式是：

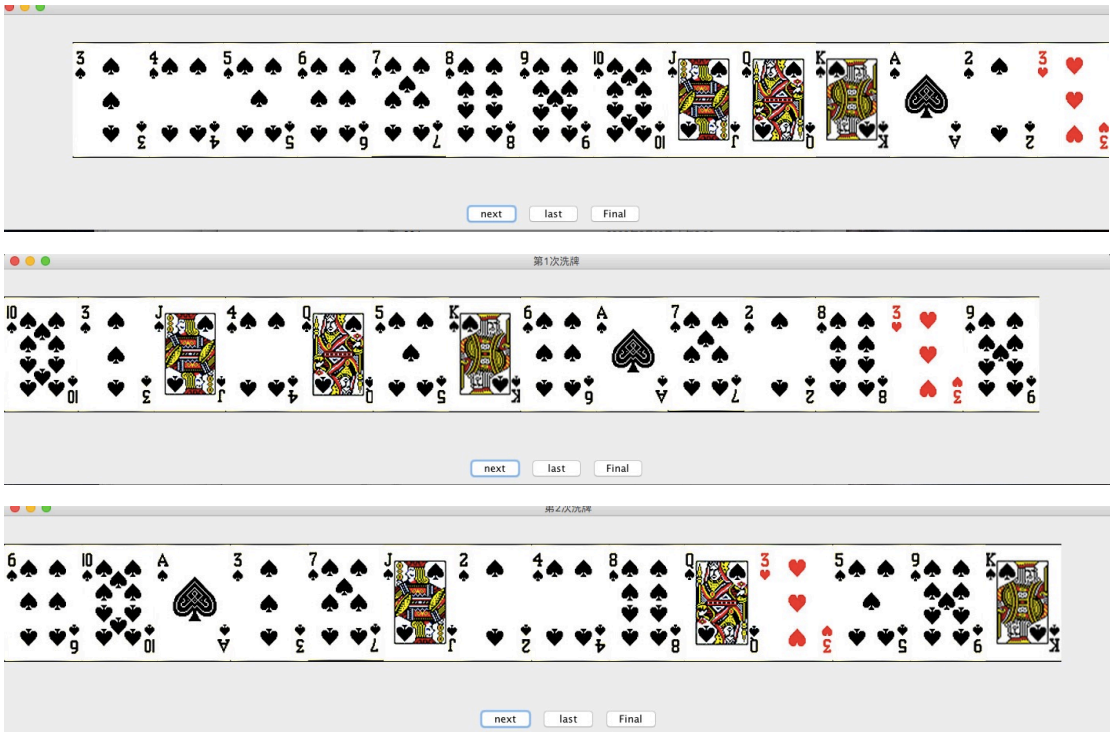
F（x）＝（2x）％（n＋1）;

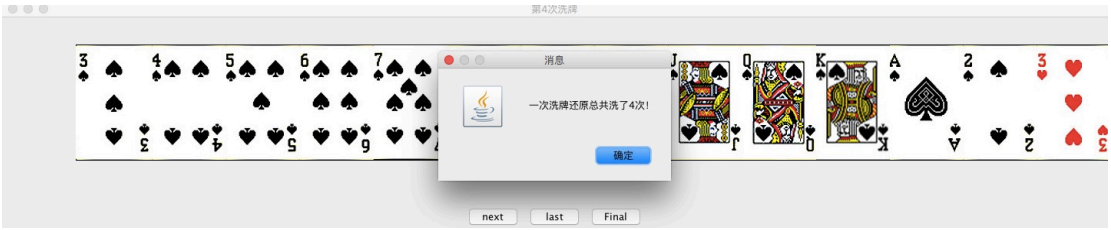
X 表示这张牌的位置，n 表示扑克牌的总数目。

但这个算法还存在一些局限，比如扑克牌的数目只能是偶数而不能是奇数。

这个算法的根本是离散数学中不相交循环的数学知

识，这样洗牌洗多少次会恢复初始状态，次数越多，也就说明这样的扑克牌越安全，应用到密码领域就是越不容易被破解。在数学知识中可以了解到应该选用选用大素数，并且这个素数的欧拉函数取值取到最大，这样就可以保证循环的次数最多。从这个算法公式可以推导得出，52 这个数字相对来说非常安全，一副 52 张牌需要洗 52 次才能还原，然而 50 张牌只需要 8 次就可以还原，这也就解释了为什么扑克牌选择 52 张牌（在外国很多扑克牌游戏都是去掉大小鬼）。





这样可以清楚地显示洗牌的过程和算法的每一步结果。

同时测试了 52 张牌和 50 张牌的理论洗牌次数：