Lab1 实验报告

191300073 杨斯凡

Lab1 的实验要求是不使用乘除法完成一个(a×b)mod m 的计算函数，并且 a×b 有可能溢出。其中 a，b 都是无符号数，则 a，b 小于2^{64}，因此 a×b 就很有可能溢出。

因为实验不能使用乘法，所以就必须用加法来实现乘法，我的方法是仿照数电课上讲过的加法器，并且在 jyy 老师的主页上也有提示，我使用两个数组来存储 a，b 的二进制形式，则在这个数组中，每个元素的索引代表着那个位置上的数字乘了 2 的多少次方，而 a 就可以表示成每个元素为 1 的位置乘 2 的索引次方，而 a×b 我也使用了一个数组储存，我是将 a 与 1 进行与运算，这样就得到了 a 最低位的值，再将 a 右移，这样就可以记录下一位。因为 a，b 我都是用数组储存，而 a×b 就可以看做两个多项式乘积，而 a×b 的每个索引上的元素就是 a 中的索引和 b 中的索引之和，而这里使用两个 for 循环就可以算出 a×b 的大小，进而把这个结果使用 SUM 数组进行存储，进而去计算余数。

而计算余数的时候，我定义了一个变量去存储余数，先把变量初始化为大于 m 的值，每一位左移求和就是乘积，而当变量大于 m 的时候，就使用变量减去 m，直到变量小于 m，再加上每一位之后左移。直到 SUM 数组到头，但是这里会有越界的问题出现，当 SUM 数组大于 64 位，那么移动的时候就有可能出现越界，我的解决方法是判断最高位是不是 1，如果是 1 的话就证明会出现一次2^{64}，因为 m 是无符号数，那么 m 的取反即-m 就是2^{64}模 m 的余数，我就将-m 加到变量中，继续循环，这就解决了越界的问题。