**湖北工业大学**

**实习实践报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | 张忆周 | **性别** | 男 | **学号** | 2210541109 |
| **分院** | 计算机学院 | **专业** | 信息安全 | | |
| **实践**  **单位** | 无 | **实践**  **岗位** | 岗位 无 | **实习**  **时间** | 2024-01-08至2024-01-19 |
| 岗位② |
| 岗位③ |
| **（一）实践主要内容及进程** （请按实践时间或实践内容顺序进行陈述。要求500字（包括图表）以上。表格不够可加页）      第一天，先将无符号型大整数的基本运算先做出来，之后加入有符号型以及一些其他的运算。第一天，先写了无符号型大整数的加减乘法，用存储每个大整数，从低位到高位存储。对于加法，采用模拟用竖式计算的方法，对于,进位为，，则;对于减法运算，与加法类似，last表示借位，，实现时要注意清除因借位产生的前导零；对于乘法，目前采用模拟竖式计算的乘法，时间复杂度较高为。    第二天，由于在除法的实现中要用到比较大整数的大小，于是加上了和，比较的实现是非常简单的，先比较位数，位数相同再从高位到低位依次比较每位的大小即可；除法也是采用模拟的方式来实现，每次被除数减去除数的倍，只需找到当前数量级下使得成立的最大的即可;取余的运算比较简单,即可；模幂运算，即计算，采用快速幂算法即可。再以上的三种运算中，都用到了大量乘法运算，由于所采用乘法运算的低效导致的。    之后的几天，在之前的基础上，添加了符号位，用表示，类型，时表示负数；用,表示两个大整数，表示运算结果。在加法运算中，ab同号时，即时，，，异号时，时,,时,。在减法运算中，时，，时，，同号时，时,，时，。除法和乘法相对比较简单，同号时，异号时为。完成这些后，测试了几个用例，发现有几个减法、除法和取余的结果不对，分析代码发现，在减法运算时,之前是基于且的情况下实现的，所以要判断当时返回；而在除法中也类似，时需返回；取余的错误是由于减法和除法的错误产生，修改之后便都正确了。    在实现了加、减、乘、除、余、模幂、求逆等操作后，便可以实现一个简单的加密算法。先选取了两个大质数和，然后算出公钥，计算，然后取一个,使得，这里选取的为，求私钥，使，即为模的逆元。如此得到公钥为，私钥为。让后对明文使用公钥加密得到密文，解密时用私钥采用同样的运算解密得到。  **（二）主要收获与体会** （请按实践时间或实践内容顺序进行陈述。要求005字（包括图表）以上。表格不够可加页）  在这段时间里，实现了大整数加、减、乘、除、余、模幂以及求逆运算，并以此为基础实现了RSA加密算法，在这过程中也让我去学习到了很多算法和数论知识。  大整数的加与乘运算，以及模幂运算由于之前有写过，所以这部分的实现是比较轻松的，减法与加法的实现类似，所以减法也是轻松实现。但是对于除法，这与前面的几种运算完全不同，所以就去看了几篇博客学习了两个大整数除法运算（不过目前网上大部分都是大整数与或型的除法，我之前也只写过这种，不过从最后的实现来看其实实现起来是差不多的，大整数除大整数的比较少可能是因为这个过程要用到大整数乘法而大部分文章都是单独实现的除法）。除法实现后取余也就简单了。再完成了基本运算后，就去实现模意义下的乘法求逆运算。最开始先去看了一些网上的文章，求逆有几种实现方法，我先去学习了第一种基于费马小定理实现的求逆运算，学习到了费马小定理，在实现了这个之后，发现这种实现方法只在模数为质数时成立，因此我去学习了第二种实现方法—基于扩展欧几里得算法的求逆元，这种方法只要该数与模数互质便可求得逆元，最后也是采用了这种方法。虽然求逆还有第三种方法，在线性的时间复杂度里求得逆元，但是这种利用了一个递推式，考虑在使用大整数时递推会用到非常大的数组，根本存不下，使用没有采用这个方法。之后测试几组数据，发现求逆结果不对，后经调试发现是由于运算过程中有负数产生，而我写的是无符号型的，便又去加了个符号位，且进行了调整使其支持有符号的大整数运算。  在实现了以上几种运算之后，便开始对其进行应用，于是基于此写了个加密算法，这里上课都讲过所以也是很简单就写出来了。不过由于RSA中有用到两个非常大的数的乘法运算（并且在加解密的过程中使用了的模幂运算，虽然使用了快速幂算法，但由于幂次方数量级过于大，所以任有大量乘法），而目前乘法运算时间复杂度为，分别为两数的数量级，所以非常慢，于是尝试的去学习了快速傅里叶变换求优化，实践的那段时间也是大部分在搞这个算法，快速傅里叶变换算法原理还是比较复杂的，有一定的前置知识需要理解，但其实现比意想中要简单，非常简短，但由于学艺不精，实现细节有些问题，运行速度并没有优化多少，并且有时运行结果会有些许偏差（可能是由于要转换为复数再转换回来出现了精度丢失，或许该去学学快速数论变换了?），所以目前并没有将其替换上去。  **（三）对实践单位的建议**  无。  **（四）实践成果**  代码链接:  https://github.com/slmhh/homework/blob/main/%E5%A4%A7%E6%95%B4%E6%95%B0%E5%BA%93/2024.1.19  .cpp  以下是主要函数截图以及运行测试截图:  **文本  描述已自动生成**  **文本  描述已自动生成**  **文本  描述已自动生成**  **文本  描述已自动生成**  **文本, 信件  描述已自动生成**  **文本  描述已自动生成**  RSA部分:  **文本  描述已自动生成**  **文本, 信件  描述已自动生成**  **运行结果: 图形用户界面, 文本  描述已自动生成** | | | | | |