Miller-Rabin 素性检验

计86 罗境佳 2018013469

运行方式

在 `src` 目录下，运行 `make run`。

实现

本次实现了一个 Miller Rabin 算法用以检测最长为 512 bit 的大数是否为素数。

大数的实现

见文件 `num.h`，定义了 `u512` 类。

采用长度为 65 的 u8 数组用以存储大整数，用 u8 类型是为了规避不同机器上字节序的影响，采用 65 是为了应对计算过程中溢出的麻烦。在该类中实现了 `+`、`-`、`%`、`&`、`|`、`>>`、`<<` 等运算符。

乘法、乘幂和除法运算用单独的函数实现。

其中，除法函数定义在 `u512::divide` 函数中，采用二进制长除法计算商以及余数。

模数乘法以及模数乘幂则定义在 `main.cpp` 中的 `mul` 函数以及 `pow` 函数。

Miller Rabin 算法的实现

见 `main.cpp` 中的 `isPrime` 函数，伪代码如下：

```
function isPrime
input: BigNumber n
output: bool is prime

calculate odd number m and k
n - 1 = m * 2 ^ k

for i in 0..TEST_ROUND:
```

```
    get random number a
    y = a^m mod n
    if y == 1 or n-1:
        说明最终能通过Fermat检验和平方根检验
        continue
    for j in 1..k:
        x = a^(m + 2j), y是上一步的x
        if x == 1 and y != 1 and y != n - 1:
            说明不能通过平方根检验
            return false
        else if x == 1 or x == n - 1:
            说明可以通过两个检验，进入下一个test round
    if x != 1:
        说明没有通过Fermat检验
        return false
return true
```

正确性

收集了 6 个测例，均为最高比特为 1 的 512 位大数。其中 4 个素数，2 个合数。最终均通过了检测。

这几个大数写在了 `prime.txt` 文件中，但读入方式实际上是从硬编码的 `string` 中读入。