密码学实验实验报告一

18374480-黄翔

2021 年 3 月 12 日

1 实验目的

1. 通过本次实验，熟悉编程环境，为后续实验做好铺垫。
2. 回顾数论的基本算法，加深对其理解，为本学期密码学课程及实验课打好基础。

2 实验环境

python 3.9.1+

3 实验内容

3.1 厄拉多塞筛法

3.1.1 算法流程

对于给定的输入 N，从 2 开始将 $\sqrt{N}$ 内的素数的倍数标记为合数，标记完后输出从 2 开始的不大于 N 的未标记的数，即为范围内全部素数

3.1.2 算法伪代码

Algorithm 1 厄拉多塞筛法

Input: $N > 0$

Output: 不大于 N 的所有素数

```
for i = 2 to N do
    flag[i] = True
```

end for
for $i = 2$ to $\lfloor\sqrt{N}\rfloor$ **do**
    if $flag[i]$ is True **then**
        $j = i * i$
        while $j <= N$ **do**
            $flag[i] = False$
            $j = j + i$
        end while
    end if
end for

3.1.3 测试样例及结果截图

测试样例及结果见附件 out 以及 output

3.1.4 总结

算法优化 伪代码中算法需要为 N 内全部数开辟数组空间，空间复杂度 $O(N)$，时间复杂度为 $O(N\log\log N)$。通过只筛取奇数，将内存占用降低一般。进一步的，采用分块筛法，只需维护小于 $\sqrt{N}$ 的素数数组以及分快数组，空间复杂度降低为 $O(\sqrt{N}+S)$

3.2 欧几里得算法

3.2.1 算法流程

先求 $p=(a,b)$。构造解 (x_n,y_n)

$$\begin{cases} x_n = \dfrac{c}{p} \\ \forall y_n \in Z \end{cases}$$

在递归的回溯过程中，利用公式：

$$\begin{cases} x_{i-1} = & y_i \\ y_{i-1} = & x_i - \lfloor a_{i-1}/b_{i-1}\rfloor \times y_i \end{cases}$$

倒推每一组 (a_i,b_i) 的解 (x_i,y_i)。最后得到 (a,b) 和原方程 $a\times x_0+b\times y_0=c$ 的 x_0,y_0

3.2.2 算法伪代码

Algorithm 2 欧几里得算法

Input: a, b

Output: $x = gcd(a, b)$

while $b \neq 0$ **do**
    $temp \leftarrow b$
    $b \leftarrow a \mod b$
    $a \leftarrow t$
end whilereturn $x \leftarrow t$

Algorithm 3 拓展欧几里得算法

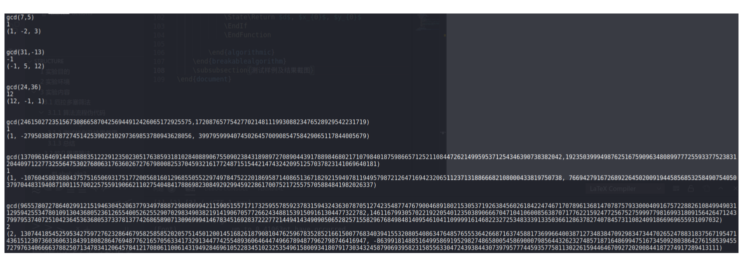
Input: a, b

Output: $gcd(a, b)$, x, y: $x \times a + y \times b = gcd(a, b)$

function ExGCD(a, b)
    if b is 0 **then**

        return $gcd(a, b) \leftarrow a$, $x \leftarrow 1$, $y \leftarrow 0$
    else
        d,x_1,y_1 = ExGCD(b, $a \mod b$)
        x_0, $y_0 = y_1$, $x_1 - [a/b] \times y_1$
        return d, x_0, y_0
    end if
end function

3.2.3 测试样例及结果截图



3.3 快速幂取模

3.3.1 算法流程

初始化 d 为 1，对 n 每次除以 2 的余数，若为 1 则 $d = d^x \mod m$，再整除 2，否则只需整除 2。$x = x^2 \mod m$，若此时 $n > 0$，继续上述流程。

3.3.2 算法伪代码

Algorithm 4 快速幂取模

Input: x, n, m

Output: $x^n \mod m$

$d \leftarrow 1$

while $n > 0$ **do**

  if $n \mod 2$ is 1 **then**

    $d \leftarrow (d^x) \mod m$

    $n \leftarrow \frac{n-1}{2}$

  else

    $n \leftarrow \frac{n}{2}$

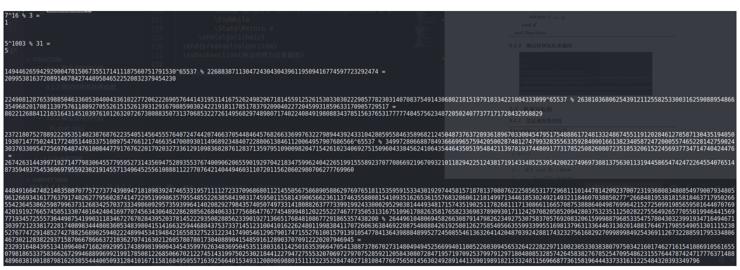
  end if

  $x \leftarrow (x \times x) \mod m$

end while

return d

3.3.3 测试样例及结果截图



3.3.4 总结

算法复杂度分析 常规模幂算法时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。快速模幂算法时间复杂度为 $O(\log n)$，空间复杂度为 $O(1)$

3.4 中国剩余定理

3.4.1 算法流程

计算 $M = m_1 \times m_2 \times \cdots \times m_k$，定义 $M_i = M/m_i$，再分别求出 M_i 在 m_i 下逆元 e_i。最终计算 $x = \Sigma(M_i \times e_i \times a_i)$

3.4.2 算法伪代码

Algorithm 5 中国剩余定理

Input: $a = [a_1, a_2, \cdots, a_k], m = [m_1, m_2, \cdots, m_k]$

Output: x

$M \leftarrow m[1] \times m[2] \times \cdots \times m[k]$

for $i \leftarrow 1$ to k **do**

  $e[i] \leftarrow inverse(\frac{M}{m[i]}, m[i])$

end for

$sum \leftarrow 0$

for $i \leftarrow 1$to k **do**

  $sum = sum + \frac{M}{m[i]} \times e[i] \times a[i]$

end forreturn $sum \mod M$

3.4.3 测试样例及结果截图

3.4.4 总结

算法复杂度 中国剩余定理将拓展欧几里得算法，模逆求解结合，是同余方程求解重要方法。时间复杂度为 $O(k)$

算法拓展 上述算法需要满足模数 m_i 互素，但是对不互素情况，同余方程组依然可能有解。以两个同余方程方程组为例

$$\begin{cases} x \equiv a_1 \mod m_1 \\ x \equiv a_2 \mod m_2 \end{cases}$$

只需要满足 $\frac{a_2-a_1}{gcd(m_1,m_2)}$ 有整数解即可。且二者有解时可合并为同余方程组，令 $d = gcd(m_1, m_2)$, $p_1 = \frac{m_1}{d}$, $p_2 = \frac{m_2}{d}$， $gcd(p_1, p_2) = 1$，根据

$$\begin{cases} k_1 \times p_1 - k_2 \times p_2 = \dfrac{a_2 - a_1}{d} \\ r_1 \times p_1 + r_2 \times p_2 = 1 \end{cases}$$

不难得到

$$x \equiv (a_1 + \frac{a_2 - a_1}{gcd(m_1, m_2)} * r_1 * m_1)\% lcm(m_1, m_2) \mod lcm(m_1, m_2)$$

对于多个同余方程，依次合并即可

3.5 素性检测算法

3.5.1 算法流程

对 N 为奇数，$N - 1 = 2^k \times q$，其中 q 为奇数，取满足 $gcd(a, N) = 1$ 的 a, 如果

$$\begin{cases} a^q \mod N \neq 1 \\ a^{2^i} \times q \mod N \neq N - 1 (\forall i = 0, 1, \cdots, k-1) \end{cases}$$

则确定 N 为合数。

3.5.2 算法伪代码

Algorithm 6 Miler-Rabin 素性检测算法

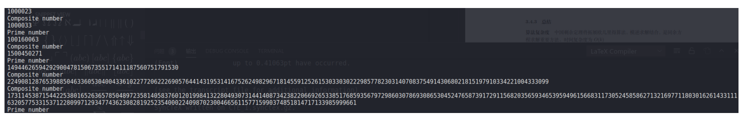
Input: n

Output:

1: get k is odd, $q : n - 1 = 2^k \times q$
2: select a randomly, and $1 < a < n - 1$
3: **if** $a^q \mod n$ is 1 **then**
4:     **return** Not Sure

```
else
    for j ← 0 to k − 1 do
        if then a^(2^j × q) mod n is n − 1
            return Not Sure
        end if
    end for
    return Compsite Num
end if
```

3.5.3 测试样例及结果截图



3.5.4 总结

素性检测可靠性 设 n 是一个奇合数，那么取值在 1 到 $n-1$ 之间的 a 至少有 75% 概率是 n 的 MIller-Rabin witness。因此，每次判断都有 75% 的可能得到 witness。若在 T 次检测中都没有找到 witness，则 n 是合数的概率为 $(25\%)^T$。$T=100$ 时都不能得到 n 是合数, 则此时 n 是合数的概率约为 10^{-60}，因此判断 n 为素数是可靠的。

4 总结

本次实验完成了厄拉多塞筛法，欧几里得算法，快速模幂算法，中国剩余定理，Miller-Rabin 素性检测算法的 python 实现。回顾了信息安全数学基础，并加深了对每个算法的理解。同时学习了算法的优化方法等。