问题求解（三）第14周作业

黄奕诚161220049

December 4, 2017

TJ Chapter 7

3

尝试了各种方法......仍然不是很懂这个密码是什么玩意(答案里的hint感觉有问题？)

7

(a) 因为$629 = 2^9 + 2^6 + 2^5 + 2^4 + 2^2 + 2^1$，所以$31^{628} \mod 3551 = 31^{2^9} \cdot 31^{2^6} \cdot 31^{2^5} \cdot 31^{2^4} \cdot 31^{2^2} \cdot 31^{2^1} \mod 3551 = 1547 \cdot 1844 \cdot 2466 \cdot 2535 \cdot 261 \cdot 961 (\mod 3551) = 2791$，所以密码为2791.

(b) $23^{47} = 23^{2^5+2^3+2^2+2^1+2^0} == 769 (\mod 2257)$，所以密码为769.

(c) $14^{13251} = 14^{2^{13}+2^{12}+2^9+2^8+2^7+2^6+2^1+2^0} = 109182 \cdot 53571 \cdot 22239 \cdot 17287 \cdot 51691 \cdot 106913 \cdot 196 \cdot 14 = 112135 (\mod 120979)$
同理$23^{13251} = 25032 (\mod 120979)$，$71^{13251} = 442 (\mod 120979)$，因此密码为112135 25032 442.

(d) $23^{781} = 4438 (\mod 45629), 15^{781} = 16332 (\mod 45629), 61^{781} = 31594 (\mod 45629)$，因此密码为4438 16332 31594.

9

(a) $2791^{1997} = 2971^{2^{10}+2^9+2^8+2^7+2^6+2^3+2^2+2^0} = 31 (\mod 3551)$，故原码为31.

(b) $34^{81} = 2014 (\mod 5893)$，故原码为2014.

(c) $112135^{27331} = 112135^{2^{14}+2^{13}+2^{11}+2^9+2^7+2^6+2^1+2^0} = 112135 \cdot 63902 \cdot 95035 \cdot 84959 \cdot 62565 \cdot 74334 \cdot 96724 \cdot 105127 = 14 (\mod 120979)$，故原码为14.

(d) $129381^{671} = 21712 (\mod 79403)$，故原码为21712.

12

当$E = 1$时，$X^E \equiv X (\mod n)$对任意n和X都成立，当$n = 1$时，原方程对任意E, X都成立，此外，解还有很多种情况，但实际上构不成对RSA密码系统的威胁，因为实际应用时会避免取边界特殊数，并且一般是大数，极少存在题中情况.

TC Chapter 31

31.7-1

因为$\Phi(319) = 10 \cdot 28 = 280$，则只要求出方程$3k = 280t + 1$的当$k$最小时的正整数解即可，解得$k = 187$，于是可得$d = 187$.
当$M = 100$时，加密后的信息为$100^{187} = 100^{(2^7+2^5+2^4+2^3+2^1+2^0)} = 122 (\mod 319)$，所以加密后得到的信息是122.

31.7-2

Proof. 首先，$ed \equiv 1 (\mod \Phi(n))$，因为$e = 3$并且$d < \Phi(n)$，又$\Phi(n) = (p-1)(q-1)$，所以有$3d = k(p-1)(q-1) + 1$，其中$k = 1$或2.得到$p + q =$

$n-\frac{3d-1}{k}+1$.由此可以在关于n的位数β的多项式时间中计算出$p+q$，用$(q+p)-p-q$代替$q-1$可以在关于β的多项式时间中计算p以及q，由此得证. □

31-2

a. *Proof.* 首先将a和b用二进制表示，则它们的长度分别为$\lg a$和$\lg b$.在进行长除法的过程中，每当计算一位的商，便要对那一位和a进行一次乘法，共操作了$\lg b$次乘法.其次用a做减法，对商中每一位数字重复这一操作，直至余数小于b，此时进行了$1+\lg q$次操作，由此根据嵌套关系，可得：该算法需要执行$O((1+\lg q)\lg b)$次位操作. □

b. *Proof.* 由(a)问可知，计算$a \mod b$所需时间为$k(1+\lg q)\lg b$，其中k为某整数.由此$\mu(a,b)-\mu(b,a \mod b)=(1+\lg a)(1+\lg b)-(1+\lg b)(1+\lg(a \mod b))$.当$a \mod b$最小时，该值最大，此时是$(1+\lg a)(1+\lg b)-(1+\lg b)=\lg a(1+\lg b)$.由(a)问可以得到这一上界，所以执行的位操作次数至多为$c(\mu(a,b)-\mu(b,a \mod b))$. □

c. *Proof.* 由(b)问知，该算法在第一次递归调用时进行最多$c(\mu(a,b)-\mu(b,a \mod b))$次位操作，在第二次进行最多$c(\mu(b,a \mod b)-\mu(a \mod b, b \mod (a \mod b)))$，依此类推.累加得到总共的位操作次数是$O(\mu(a,b))$.当输入两个$\beta$位数时，需要执行的位操作次数为$O(1+\beta)(1+\beta)=O(\beta^2)$.得证. □

31-3

a. *Proof.* 由27.1节的FIB算法可知，该算法运行时间满足

$$T(n)=T(n-1)+T(n-2)+\Theta(1)$$

(a) 首先证明$T(n)=O(F_n)$：假设对任意的$k<n$有$T(k)\le cF_k-c_1k$，则有$T(n)=T(n-1)+T(n-2)+\Theta(1)\le cF_{n-1}-c_1(n-1)+cF_{n-2}-c_1(n-2)=cF_n-2c_1n+3c_1\le cF_n-c_2n$，由此可得$T(n)=O(F_n)$.

(b) 其次证明$T(n)=\Omega(F_n)$，假设对任意的$k<n$有$T(k)\ge cF_k+c_1k$，则有$T(n)=T(n-1)+T(n-2)\ge cF_{n-1}+c_1(n-1)+cF_{n-2}+c_1(n-2)=cF_n+2c_1n-3c_1\ge cF_n+c_2n$，由此可得$T(n)=\Omega(F_n)$.

综上可知，该算法运行时间为关于F_n的多项式时间. □

b. 记忆法算法如下：

Algorithm 1 FIB-MEMORY(n)

```
F_1 → 1, F_2 → 1
for i → 3 to n do
    F_i → F_{i-1} + F_{i-2}
end for
return F_n
```

c. 经过尝试可以发现

$$\begin{bmatrix}0 & 1\\1 & 1\end{bmatrix}^k=\begin{bmatrix}F_{k-1} & F_k\\F_k & F_{k+1}\end{bmatrix}$$

假设一次二阶矩阵的乘法运算时间为$O(1)$.由此可以得到如下运行时间为$O(\lg n)$的算法(使用了矩阵快速幂方法)：算法返回的$T[0,1]$的元素即为矩

Algorithm 2 FIB-MATRIX(A,n)

```
T is the unit 2 × 2 matrix
while n ≠ 0 do
    if n is even then
        T → T * A
    end if
    n → n/2
    A → A * A
end while
return T[0,1]
```

阵T右上角元素，即F_k.

d. 在对运行时间的规定修改后，
方法(a)中，$T(1)$为1，$T(0)$为0，需要的时间为$\Theta(2^n)$.
方法(b)中，运行时间为$\Theta(n^2)$.
方法(c)中，运行时间为$\Theta(n\lg n)$.