第1题：

(a): $f=\Theta(g)$　(b): $f=O(g)$　(c): $f=\Theta(g)$　(d): $f=\Theta(g)$　(e): $f=\Theta(g)$　(f): $f=\Theta(g)$　(g): $f=\Omega(g)$　(h): $f=\Omega(g)$　(i): $f=\Omega(g)$　(j): $f=\Omega(g)$　(k): $5^{\log_2 n}=(2^{\log_2 5})^{\log_2 n}=n^{\log_2 5}$, $f=O(g)$.　(l): $f=O(g)$　(m): $f=O(g)$　(n): $f=\Theta(g)$　(o): $f=\Omega(g)$　(p): $f=O(g)$

第2题：

(a): 设 $g(n)=\begin{cases} n+1, & c=1 \\ \frac{1-c^{n+1}}{1-c}, & c\neq 1 \end{cases}$

取 $f(n)=1$，$\because \frac{f(n)}{g(n)}=\frac{1-c}{1-c^{n+1}}\leq\frac{1-c}{1-c}=1$

$\therefore f(n)=O(g)$.

$\because \frac{g(n)}{f(n)}=\frac{1-c^{n+1}}{1-c}\leq\frac{1}{1-c}$

$\therefore g(n)=O(f(n))$

$\therefore f=\Theta(1)$

(b): 取 $f(n)=n$

$\because \frac{f(n)}{g(n)}=\frac{n}{n+1}\leq 1$.

$\frac{g(n)}{f(n)}=\frac{n+1}{n}=1+\frac{1}{n}<2$.

$\therefore g(n)=\Theta(n)$

(c): 取 $f(n)=c^n$,

$\frac{f(n)}{g(n)}=\frac{c^n}{\frac{c^{n+1}-1}{c-1}}=\frac{c^{n+1}-c^n}{c^{n+1}-1}\leq\frac{c^{n+1}-1}{c^{n+1}-1}=1$

$\frac{g(n)}{f(n)}=\frac{c^{n+1}-1}{c^{n+1}-c^n}=\frac{c-\frac{1}{c^n}}{c-1}<\frac{c}{c-1}$

$\therefore g(n)=\Theta(c^n)$

1-14:

解：使用了课中提到的矩阵乘法，进行 $T(n)$ 的计算。

并且在算乘方 $\begin{pmatrix}0&1\\1&1\end{pmatrix}^n$ 后对每一位取对 p 取模

即 $\begin{pmatrix}F_n \bmod p\\ F_{n+1} \bmod p\end{pmatrix} = \left(\begin{bmatrix}0&1\\1&1\end{bmatrix}^n \bmod p\right)\begin{pmatrix}0&1\\1&1\end{pmatrix} \bmod p \begin{pmatrix}F_0\\F_1\end{pmatrix}$

由课上知识知 mod 运算为 $O(N^2)$，N 为 p 二进制位数

∴ 总时间复杂度（利用了快速幂）$= O(N^2 \cdot \log n)$

$= O((\log^2 p)(\log n))$

∴ 显然小于课上提到的 $O(2^n)$ 和 $O(n)$ 的算法

1-20:

解：① $20 \bmod 79$

∵ 20 与 79 互质

∴ $20 \bmod 79$ 的模倒数（逆元）存在。

∴ 根据拓展欧几里得算法：

```
F(a,b):
{ if (b==0) return (1,0,a);
  (x',y',d) = F(b, a mod b);
  return (y', x'-(a/b)y', d);
}
```

则由 $ax+by=1$，即 $20x+79y=1$，（x 为逆元）

由程序可得：$20 \bmod 79$ 的逆元（模倒数）为 4。

即：$20^{-1} \equiv 4 \bmod 79$（取 $x=4, y=-1$）

② $3 \bmod 62$

∵ 3 与 62 互素

∴ 由拓展欧几里得可得：$3x+62y=1$（x 为逆）

$3 \bmod 62$ 的逆元为：21

即 $3^{-1} \equiv 21 \bmod 62$（取 $x=21, y=-1$）

③ ⇔ 21 mod 91

⇔ gcd(21, 91) = 7 ≠ 1

∴ 21与91不互质

∴ 不存在逆元.

④ ⇔ 5 mod 23

⇔ 5与23互质

∴ 逆元存在.

∴ 由欧几里得算法可得: $5x+23y=1$,

∴ 可得: $5^{-1} \bmod 23 = 14$ (取 $x=14, y=-3$)

(-3) ⇔ ⑤

解: 欲计算 $N!$ 的近似位数, 则为计算 $\log N!$ 的Θ表示.

⇔ $\frac{N!}{N^N} = \frac{N(N-1)\cdots 2\cdot 1}{N\cdot N\cdots N}$ (N个)

⇔ $\frac{N!}{N^N} = \frac{N(N-1)\cdots 2\cdot 1}{N^N} \le 1$

⇔ $\log N! \le N\log N$

⇔ $\frac{N!}{(\frac{N}{2})^{\frac{N}{2}}} = \frac{N\cdots 2\cdot 1}{(\frac{N}{2})\cdots(\frac{N}{2})} = \left((\lfloor\frac{N}{2}\rfloor-1)\cdots 2\cdot 1\right)\frac{N\cdots\lfloor\frac{N}{2}\rfloor}{\lfloor\frac{N}{2}\rfloor\cdots\lfloor\frac{N}{2}\rfloor} \ge 1$ ($\frac{N}{2}$个)

∴ $\log N! \ge \frac{N}{2}\log\frac{N}{2}$

∴ ~~Otto~~ $\log N! = O(N\log N)$, $\log N! = \Omega(\frac{N}{2}\log\frac{N}{2}) = \Omega(N\log N)$

∴ $\log N! = \Theta(N\log N)$.

∵ N为n位二进制 ∴ $\log N = n$, $N = 2^n$.

∴ $N!$ 位数Θ表示为 $= \Theta(N\log N) = \Theta(n2^n)$

~~函数：T(int N)~~

~~{ for (i=1; i<=N; ++i)~~

(b) 函数：T(N)：

{ int ans=1;

for(i=1; i<=N; ++i) ans *= i;

return ans; }.

时间复杂度分析：~~对于 O 分析，可省~~

$T(n)=O\left(\sum_{i=1}^{n}(i\log i)\cdot\log i\right)=O\left(\sum_{i=1}^{n} i\log^2 i\right)\le O(N^2\log^2 N)$

其中 i 为遍历到前数字，log 为 i 的进制位数。

~~= O 分析可省去常数及低次项~~

~~∴ $T(n)=O(N\log^2 N)=O(n^2 2^n)$ (n 为 N 的进制位数)~~

∴ $T(n)=O(N^2\log^2 N)=O(n^2 4^n)$

1-3.

(a) 证明：由扩展欧几里得算法，$ax+py=1$

⇒ 此 a 以自身为逆

∴ $a^2+py=1$ 且 $1\le a<p$，则 y 一定为负非正数

⇒ $a=\sqrt{1-py}$

⇒ y=0 显然成立，y=-(p-1) 成立，即 a=1 或 a=p-1 都成立

∴ 满足 $-p<y\le 0$ 且 a 为 [1,p) 范围内的 y

对应的 a 均以自身为逆，且~~为~~包含 a=1 与 a=p-1

(b) 证明：由逆元定义可知，

$ax\equiv 1 \pmod p$，其中，x 为 a mod p 的逆元

由(a)可知 1 与 (p-1) 逆元为本~~题~~身

对于 [1, p-1] 中其他数，对 $\forall a\in[2,p-2]$，

a 与逆元 x 满足 $ax\equiv 1 \pmod p$

若a逆元∈[1, p-1]，则称若a逆元>p，则：由 $ax \equiv a(x-kp) \pmod p$

∴ a逆元x仍可映射到[1, p-1]范围内，且不重复（因为若重复，则x逆元

∴ 有多个，与a逆元矛盾）

∵ p为奇数 ∴ p-1 共有偶数个，∴ 可以配对成 $\frac{p-1}{2}$ 对逆元对

∵ 若 $x \equiv x' \pmod p$，$y \equiv y' \pmod p$，则 $xy \equiv x'y' \pmod p$

∴ $(p-1)! = 1 \cdots \cdots (p-1) \equiv (p-1) \equiv -1 \pmod p$ ∴ $(p-1)! \equiv -1 \pmod p$ 成立

(c) 假设N不是素数时仍有 $(N-1)! \equiv -1 \pmod N$ 成立。

∵ $(N-1)! = kN-1$

∴ $d = gcd((N-1)!, N) > 1$.

∴ 取 $(N-1)! = k_1 d$，$N = k_2 d$

∴ $k_1 d = k_2 d - 1$

∴ $(k_1 - k_2)d = -1$ ①

∵ k_1, k_2 均为整数，$d > 1$，

∴ ①不可能成立

∴ 命题：当N不是素数时 $(N-1)! \not\equiv -1 \pmod N$

(d) 由题(3)知，计算N!时间复杂度为 $O(n2^n)$，

而费马小定理计算 a^{p-1} 只需 $O(n^3)$（n为N位数），

多项式时间优于指数时间，故Wilson定理在性能

方面优于费马定理，故不采用.