Ch1

1.1

除数为 a，被除数为 b，二者的关系可以描述为 $a=mb+r$，不断进行迭代可得:

$a=m_0*b+r_2$,

$n=m_1*r_2+r_3$,

$r_2=m_2*r_3+r_4$,

…

$r_{n-2}=m_{n-2}*r_{n-1}+r_n$

$r_{n-1}=m_{n-1}*r_n$

分析易知: $m_{n-1}\geq 2$, m_0, m_1, …, $m_{n-2}\geq 1$, $r_2>r_3>\ldots>r_n$。因为 r_{n-1} 模 r_n 的余数为 0，因此 $r_n\geq 1$。递推往上有，$r_{n-1}\geq 2$, $r_{n-2}\geq 3$, $r_{n-3}\geq 5$, ⋯ 我们猜测这种关系符合斐波那契数列，下用归纳法证。已知 $r_n\geq F_2$, $r_{n-1}\geq F_3$, 则 $r_{n-2}=m_{n-2}*r_{n-1}+r_n\geq F_4$。当 $r_{n-j}\geq F_{2+j}$, $r_{n-j-1}\geq F_{3+j}$, 则 $r_{n-j-2}=m_{n-j-2}*r_{n-j-1}+r_{n-j}\geq r_{n-j-1}+r_{n-j}\geq F_{2+j} + F_{3+j} = F_{4+j}$，满足斐波那契数列的规律。假设 $a>b$，根据通项公式，$b\geq F_{n+1}>\frac{1}{\sqrt{5}}*(\frac{1+\sqrt{5}}{2})^{n+1}$，两边取对数有$\log_2 b>(n+1)\log_2\left(\frac{1+\sqrt{5}}{2}\right)-\log_2\sqrt{5}>\frac{n+1}{5}$, $n<5\log_2 b-1$，因此求余的次数至多为 $min(5\log_2 a-1,5\log_2 b-1)$, 赋值的次数至多为 $min(10\log_2 a-2,10\log_2 b-2)$，时间复杂度为 $O(min(\log a, \log b))$。

1.2

(1)整体而言，算法是从 2 到 n 的循环，第 2、3、7 步的赋值操作不影响算法的有穷性，进一步考虑第 4-6 步是否能停止。这里内层循环是对有序数组进行位置调整，至多将 j-1 个元素移动一遍($0\leq j\leq i-1$, $2\leq i\leq n$, i, $j\in\mathbf{N}$)，满足有穷性。综上，算法必然停止。

(2)对内层循环：

循环不变量：数组 $A[1:j]$中元素的有序性。

初始：将 $A[j]$向右移一位，原始数组 $A[1:j]$中元素先后关系未变，仍然有序

循环：不断将 $A[1:j]$中的元素向右移动，原始数组 $A[1:j]$中元素先后关系未变，仍然有序。

终止：j=0 或 $A[j]\leq$key，数组 $A[1:j]$变为了数组 $A[1:j+1]$，中间留了一个空等 key 插入，原始数组 $A[1:j]$中所有元素仍然有序。

对外层循环：

循环不变量：数组 $A[1:n]$的有序性。

初始：i=2，$A[1]$来自输入且有序。

循环：i>2，$A[1, 2, \ldots, i-1]$来自输入且有序，内层循环执行一遍，将 $A[i]$加入数组 A 中，$A[1, 2, \cdots, i-1, i]$仍然来自输入且有序。

终止：$i=n+1$，算法停止，$A[1, 2, \ldots, n]$来自输入且有序，算法正确性得证。

(3)最坏情况：每次插入需要和 $A[1:i-1]$所有元素比较，比较次数为 $1+2+\cdots+n-1 = \frac{n*(n-1)}{2}$，赋值次数外层为 $3*(n-1)$，内层为 $2*\frac{n*(n-1)}{2} = n*(n-1)$，总和为$n^2+2n-3$。

最好情况：每次插入仅和 $A[i-1]$比较，比较次数为 n-1，赋值次数为$(3+2)*(n-1)=5n-5$。

平均情况：对比较次数和赋值次数求离散期望，由于服从均匀分布，可以直接对其求和再求平均。设插入第 j 个元素时，其插入位置为 k，$1\leq k\leq j$，需要比较 $j-k+1$ 次，故平均比较次数为$\frac{1}{j}\sum_{k=1}^{j}(j-k+1) = \frac{j+1}{2}$。插入 n 个元素，总平均比较次数为$\sum_{j=2}^{n}\frac{j+1}{2} = n^2/4+3n/4+1$。因此，总平均赋值次数为 $2*(n^2/4+3n/4+1)+3*(n-1) = n^2/2+9n/2-1$

1.4

输入规模：如果用存储空间大小、bit 来刻画，当输入正整数 n，对应的规模为$\log_2 n$

基本操作个数：根据算法范围，至多为$\sqrt{n}$，用输入规模表示为$2^{1/2*\log n}$

算法复杂度为 $O(n^{1/2})=O(2^{1/2*\log n})$，因此该算法是指数时间算法

1.5

输入规模：n

基本操作个数：算法是对数组第 2 到 n 个元素两两求和递推，因此操作个数为 n-1，算法复杂度为 $O(n)$，因此该算法是多项式时间算法。

Ch2

(1) 由定义，$o(g(n))$的集合可以视为$\{f_1(n)|\forall c_1, \exists n_1, \forall n>n_1, f_1(n)<c_1g(n)\}$，$\omega(g(n))$的集合可以视为$\{f_2(n)|\forall c_2, \exists n_2, \forall n>n_2, f_2(n)>c_2g(n)\}$，不妨取 $n_1=n_2$。下反证，若二者交集不为空，则必存在$f_1(n)$中的元素属于$f_2(n)$，根据$\omega(g(n))$的定义，$f_1(n)$可以大于 $c_2g(n)$，又因为 c_1 和 c_2 都可任取，则$f_1(n)$可以大于 $c_1g(n)$，与 $o(g(n))$的定义矛盾，故命题得证。

(2) 根据Θ定义，命题等价于证明$\exists c_1, c_2>0, n_0, \forall n>n_0, c_1[f(n)+g(n)]\le max[f(n), g(n)]\le c_2[f(n)+g(n)]$。因为当 n 足够大时，由渐近函数定义，$f(n)$和 $g(n)$均为正值函数，则$[f(n)+g(n)]/2\le max[f(n), g(n)]$，$max[f(n), g(n)]\le f(n)+g(n)$。取 $c_1=1/2$, $c_2=1$, n_0 足够大即满足Θ条件，命题得证。

(3) 根据Θ定义，命题等价于证明$\exists c_1, c_2>0, n_0, \forall n>n_0, c_1 n\log n\le \log(n!)\le c_2 n\log n$，而 $\log(n!)=\log(n)+\log(n-1)+\cdots+\log(2)+\log(1)$。取 $n_0=1$，$c_2=1$，当 $n>1$ 时，$\log(n)+\log(n-1)+\cdots+\log(2)+\log(1)\le n\log n$，$\log(n!)=O(n\log n)$得证。取 $n_0=1$，$c_1=\varepsilon$，ε是一个很小但大于 0 的常数，当 $n>1$ 时，$\varepsilon n\log n\le\log(n)+\log(n-1)+\cdots+\log(2)+\log(1)$，$\log(n!)=\Omega(n\log n)$得证。综上，命题成立。

(4) $T(n)=T(\frac{3}{10}n)+5n=T((\frac{3}{10})^2n)+5n+5*\frac{3}{10}n=\ldots=T((\frac{3}{10})^kn)+5n+5*\frac{3}{10}n+\cdots+5*(\frac{3}{10})^{k-1}n=T((\frac{3}{10})^kn)+5n/\frac{7}{10}*(1-(\frac{3}{10})^k)$。令 $n=(\frac{10}{3})^k$，$T(n)=T(1)+\frac{10}{7}*5n*(1-\frac{1}{n})=\Theta(n)$

(5) $T(n)=T(\lceil n/2\rceil)+1= T(\lceil n/2^2\rceil)+1+1=\cdots= T(\lceil n/2^k\rceil)+k$。令 $n=2^k$，$T(n)=T(1)+\log n=\Theta(\log n)$

(6) 设 $n=2^m$，则 $T(2^m)=3T(2^{m/2})+m$。令 $T(2^m)=S(m)$，有 $S(m)=3S(m/2)+m$。使用 master 定理，$m^{\log_b a}=m^{\log_2 3}\approx m^{1.58}$, $f(m)=m=o(m^{1.58-\varepsilon})$，$\varepsilon=0.5$。所以 $\mathrm{S(m)}=\Theta(m^{\log_2 3})$，$T(n)= \Theta(\log n^{\log_2 3})$

(7) $T(n)=3T(n-1)+ 2^n=3T(n-2)+ 2^n+2^{n-1}=\cdots=3T(n-k)+ 2^n+2^{n-1}+\cdots+2^{n-k}=3T(n-k)+2^{n+1}-2^{n-k}$。令 $n=k$，$T(n)=3T(0)+ 2^{n+1}-1=\Theta(2^n)$

(8) 猜测 $T(n)=O(n^2)$，需证 $T(n)\le cn^2$。带入原式，$T(n)\le cn^2/4+9cn^2/16+n=13cn^2/16+n$。当 $c\geqslant 16/3$，$T(n)\le cn^2$，命题得证。

(9) 使用 master 定理，$n^{\log_b a}=n^{\log_4 2}=n^{0.5}$，$\mathrm{f(n)}=n^{0.5}=\Theta(n^{0.5})$。因此 $T(n)=n^{0.5}\log n$

(10) $T(n)=T(\lfloor n/2\rfloor)+n^3= T(\lfloor n/2^2\rfloor)+n^3+\lfloor n/2\rfloor^3=\cdots= T(\lfloor n/2^k\rfloor)+n^3+\lfloor n/2\rfloor^3+\cdots+\lfloor n/2^{k-1}\rfloor^3\le T(\lfloor n/2^k\rfloor)+n^3+(n/2)^3+\cdots+(n/2^{k-1})^3=T(\lfloor n/2^k\rfloor)+8n^3/7*(1-8^{1-k})$。令 $n=2^k$，$T(n)=\mathrm{T}(1)+8n^3/7*(1-8/n^3)=\Theta(n^3)$

(11) 使用 master 定理，$n^{\log_b a}=n^{\log_3 5}\approx n^{1.46}$，$f(n)=n=o(n^{1.46-\varepsilon})$，$\varepsilon=0.4$，所以 $\mathrm{T(n)}=\Theta(n^{log_3 5})$

(12) $T(n)=4T(n/2)+n^2\log n=4[4\mathrm{T}(n/4)+(n/2)^2\log(n/2)]+n^2\log n=\cdots=4^kT(n/2^k)+kn^2\log n-(k-1+k-2+\cdots+1+0)n^2=4^kT(n/2^k)+kn^2\log n-n^2k(k-1)/2$。令 $n=2^k$，$T(n)=n^2\mathrm{T}(1)+1/2(n\log n)^2+n^2\log n/2=\Theta((n\log n)^2)$

(13) 设 $n=2^m$，$\mathrm{T}(2^m)=2^{m/2}(2^{m/2})+2^m$,等式两边同除以 2^m，$T(2^m)/2^m=(2^{m/2})/2^{m/2}+1$。令 $S(m)=T(2^m)/2^m$，原始等式变为 $S(m)= S(m/2)+1$。由 master 定理，$S(m)=\Theta(\log m)$,$T(2^m)=\Theta(2^m\log m)$，所以 $T(n)=\Theta(n\log(\log n))$