**作业1：习题1.1 6, 7；习题1.2 4；习题2.1 7；习题2.2 3 a. c. e.**

**习题1.1**

**6．提示：比较除法的次数**

a．gcd(31415, 14142)

= gcd(14142, 3131)

= gcd(3131, 1618)

= gcd(1618, 1513)

= gcd(1513, 105)

= gcd(105, 43)

= gcd(43, 19)

= gcd(19, 5)

= gcd(5, 4)

= gcd(4, 1)

= gcd(1, 0)

=1

故：31415、14142的最大公约数是1，即31415和14142互质。

b．① Euclid算法执行除法11次（如a所示）；

② 连续整数检测算法是从14142开始的，将检查14142轮（因为两个数互质）

每一轮执行除法1次或2次（当被检查数是31415、14142的因子时，执行2次除法）

即，除法的总次数是：1×14142～2×14142

③ 故，Euclid算法比连续整数检测算法快：

1×14142/11～2×14142/11≈1300～2600倍。

**7．提示：特意选取m＜n的两个数，执行算法一次迭代**

①对任意0≤m＜n有m=0×n+m

由Euclid算法，应有m🡨n、n🡨m，即gcd(m,n)=gcd(n,m)

故当m＜n时，Euclid算法的首次迭代将交换m、n；

②由gcd(m,n)=gcd(n,m mod n)可知，n比m mod n大

即除了首次的可能交换外，后面的迭代中第一个数总是比第二个数大。

**习题1.2**

**4．提示：算法应该对系数的所有可能值都正确工作，即使系数＝0**

//求方程ax2+bx+c=0的实根的算法

Quadratic( a, b, c )

//输入：实系数a,b,c

//输出：实根或者无解信息

if a≠0

D←b\*b-4\*a\*c

If D>0 //两个不同实根

temp←2\*a

x1←(-b+sqrt(D))/temp

x2←(-b-sqrt(D))/temp

return x1, x2

else if D=0 //两个相同实根

return –b/(2\*a)

else //D＜0

return “无实数根”

else //a=0 退化为1元1次方程 bx+c=0

if b≠0

return –c/b

else //a=b=0 退化为1元0次方程 c=0

if c=0

return “系数全0” //无穷个实数解

else

return “无实数根” //方程无解

**习题2.1**

**7．**a．，解1000个方程的运行时间，其中常数c是乘法操作执行1次所需的时间。

同理，高斯消元法解500个方程的运行时间，则有：



故：解1000个方程的运行时间比解500个方程多8倍

b．令在老计算机上算法运行时间=

则1000倍速度的新计算机的运行时间=

∵时间不变，即T(n1)=T(n2)，则有，即n2/n1≈10

故：1000倍速度的新计算机在相同时间内求解的问题规模大约增加了10倍

**习题2.2**

**3.对给定的函数进行化简。挑选出一个能够定义它们的增长次数的项**

a．推测（忽略低次项，忽略常数）：

证明：

∵

∴

c．推测（分2部分）：



=

证明：① 第1部分：

∴2*n*lg(n+2)2=Θ(*n*lg*n*)

② 第2部分：

∵(n+2)2lg(n/2)=(n2+4n+4)(lg*n*-1)≤(n2+4n2+n2)(lg*n*-1)=6n2(lg*n*-1)≤6n2lg*n*

即(n+2)2lg(n/2)=O(n2lg*n*)，n＞1

又∵(n+2)2lg(n/2)=(n2+4n+4)(lg*n*-1)≥n2(lg*n*-1)，n＞1

而

即n2(lg*n*-1)=Θ(n2lg*n*)，则(n+2)2lg(n/2)=Ω(n2lg*n*)

∴(n+2)2lg(n/2)=Θ(n2lg*n*)

③ 综合①②，2nlg(n+2)2+(n+2)2lg(n/2)=Θ(*n*lg*n*)+Θ(n2lg*n*)= Θ(*n*2lg*n*)

e．推测：=Θ(log2 *n*)

证明：∵∴，即=O(log2 *n*)

又＞log2 *n* - 1≥log2 *n* - = （若n≥4），即=Ω(log2 *n*)

综上所述：=Θ(log2 *n*)