面向对象程序设计基础 第二次作业

徐浩博 2020010108

模型部分

- 首先计算最大公约数（gcd），我们采用辗转相除法。

  假设两正整数为 a，b.（不妨设 $a\geqslant b$），且有 $a=pb+r$（$p\geqslant 1$），下分情况讨论：

  ① $r=0$ 时，则 $b\mid a$，此种情况说明，a，b 的最大公约数为 b.

  ② $r\neq 0$ 时，设 m 为 a，b 的最大公约数，则假设 $a=a_1m$，$b=b_1m$，其中$(a_1,b_1)=1$.

  $\Rightarrow a=pb+r=pb_1m+r=a_1m$

  $\Rightarrow r=(pb_1-a_1)m$

  而 $pb_1-a_1\geqslant 0$，这是因为 $p\geqslant 1$ 且 $a_1\geqslant b_1$，特别地，$r\neq 0$，则 $pb_1-a_1\geqslant 1$.

  现已知道 r 与 b 存在公约数 m，下面证明 pb_1-a_1 与 b_1 最大公约数为 1.

  设 n 为 pb_1-a_1 与 b_1 最大公约数，则假设 $pb_1-a_1=s\cdot n$，$b_1=t\cdot n$，有 $n\mid b_1$，则有 $n\mid pb_1-a_1$，于是 $n\mid a_1$，则 n 为 a_1，b_1 公约数，考虑到$(a_1,b_1)=1$，则 $n=1$.

  综上，r 与 b 存在公约数 m 且不存在更大的公约数，则 m 为 r，b 的最大公约数，则 $gcd(a,b)=gcd(b,r)=m$.

  对于②，计算 a，b 的最大公约数时，只需计算 b，r 的最大公约数（$r<a$），与此类似，a，b 每一步都会有一个数递减，经过有限步，必然会满足①的条件.

  以上为辗转相除法的证明部分，为了实现该方法，可采用函数递归的方式模拟循环，每一次调用函数 gcd(a,b)，返回 gcd(b,a%b)的值，边界条件是 a%b==0（此时满足证明中的①条件）．通过这种方法，即可完成对 gcd 的计算.

- 其次,我们再来计算最小公倍数(lcm)．设 m 为 a,b 的最大公约数,则假设 $a=a_1m$，$b=b_1m$，其中$(a_1,b_1)=1$. 设 n 是 a，b 的最小公倍数.

  为了满足 $a\mid n$，有 $m\mid n$. 不妨设 $n=n_1m$，则由 $a\mid n$，有 $a_1m\mid n_1m$，从而 $a_1\mid n_1$；同理 $b_1\mid n_1$. 设 $n_1=pa_1$，则 $b_1\mid pa_1$，则存在 q 使 $qb_1=pa_1$. b_1 中不含 a_1 的质因数，则 q 包含 a_1 的所有质因数，从而 $a_1\mid q$，从而 $q\geqslant a_1$，从而 $n=n_1m=pa_1m=qb_1m\geqslant a_1b_1m$. 考虑到 a_1b_1m 恰为 $a=a_1m$ 与 $b=b_1m$ 的公倍数，则 a_1b_1m 为 a，b 的 lcm，从而有 $lcm=a\cdot b/m=a\cdot b/gcd$.

- 为了使用面向对象的程序设计解决该问题，除了主函数所在的 CP_IntegerCalculationMain.cpp 文件外，我共编写了 4 个模块，其名称与主要功能分别为：

  CP_IntegerInput：调用以实现对正整数的读入功能.

  CP_LCMGCDCalculation：调用以实现对 lcm/gcd 的计算功能.

  CP_Time：调用以实现对开始、结束时间的测量及计算.

  CP_TimeApplication：调用以解决整个问题，主要通过以上模块的调用，实现 gcd/lcm 的计算，同时得出运算时间.

验证部分

等价类划分

①输入的不是正整数.

②输入的某个正整数超过了 int 范围.

③输入的正整数虽未超 int 范围，但计算出的 lcm 超过了 int 范围.
④输入的正整数 gcd 与 lcm 均能计算出正确结果.

案例选取

①输入的不是正整数：(a, 1.2)
②输入的某个正整数超过了 int 范围. (2,3000000000)
③输入的正整数虽未超 int 范围，但计算出的 lcm 超过了 int 范围. (6666666,6666667)
④输入的正整数 gcd 与 lcm 均能计算出正确结果. (3,4) (4,32768)(36,27) (60000006,90000009)

测试结果

输入	gcd	gcd 计算时间	lcm	lcm 计算时间
(a, 1.2)	输入非法	N/A	输入非法	N/A
(2,3000000000)	输入超出范围	N/A	输入超出范围	N/A
(6666666,6666667)	1	0ms	结果超出范围	N/A
(3,4)	1	0ms	12	0ms
(4, 32768)	4	0ms	32768	0ms
(36,27)	9	0ms	108	0ms
(60000006,90000009)	10000001	0ms	1800000018	0ms