

程序报告2

Charles Shen

2018 年 3 月 8 日

1 问题分析

读入两个正整数，输出其最大公约数和最小公倍数，验证结果正确性并输出时间。这个问题可以拆分为五个模块：读入两个正整数，计算最大公约数和最小公倍数，计时，随机样例生成和验证，输出。

2 程序实现

读入正整数 这个模块我调用了上次作业中加入代码构件库的判断并读入正整数模块(cp.integer.cpp和.h)，重复2次执行。由于上次作业中已经验证其正确性，这次不再验证。

计算最大公约数和最小公倍数 这个模块我用一个类cp_two_integers实现（具体代码在同名.h和.cpp文件中，已加入代码构件库）。

算法思想：求最大公约数 GCD 用辗转相除法，最小公倍数 LCM 则在求出最大公约数 GCD 后用公式 $LCM = a * b / GCD$ 求出。求最大公约数代码如下：

```
//事先设定好a>b
while (b>0) //当小数b=0时，停止，a即为最小公倍数。
{
    int temp = a % b;
    a = b; b = temp; //如果b不为0，则GCD(a,b)=GCD(b,a%b),可用这个公式递归求解。
}
greatest_common_divisor = a;
```

图 1: 求最小公倍数和最大公约数代码

程序验证：验证了五组数据：相等，a整除b，b整除a，一般情况，大数。验证结果如下。

```

The first number is 12
The second number is 8
Their greatest common divisor is 4
Their least common multiple is 24

The first number is 1402
The second number is 1402
Their greatest common divisor is 1402
Their least common multiple is 1402

The first number is 128
The second number is 64
Their greatest common divisor is 64
Their least common multiple is 128

The first number is 64
The second number is 128
Their greatest common divisor is 64
Their least common multiple is 128

The first number is 2017013569
The second number is 1830985851
Their greatest common divisor is 1
Their least common multiple is 3693123306114012219

```

图 2: 测试数据及结果

计时 这个模块我参考了课件里的程序，建立一个类cp_time_by_clock(代码在同名.cpp和.h文件里，已加入代码构件库),存储开始时间，结束时间。用clock函数获取开始时间和结束时间，并且借助CLOCKS_PER_SEC求出具体时间。

验证 这个模块基本思路是按照最大公约数和最小公倍数定义。当两者均为正整数，并且最大公约数为公约数，最小公倍数为公倍数，没有比最大公约数大的公约数，没有比最小公倍数小的公倍数才成立。验证几万组机器生成的随机数据后再输出结果。具体代码如下：

```

if (a <= 0 || b <= 0) return 0; //判断正整数
if (a%greatest_common_divisor!=0 || b%greatest_common_divisor!=0
    || least_common_multiple%a!=0 || least_common_multiple%b!=0) //判断是否为公约，公倍数
for (i = greatest_common_divisor+1; i <= a; i++)
{
    if (a%i == 0 && b%i == 0) return 0; //判断有没有比它大的公约数
}
for (j = b; j < least_common_multiple; j++)
{
    if (j%a == 0 && j%b == 0) return 0; //判断有没有比它小的公倍数
}
return 1;

```

图 3: 程序自动验证代码

输出 输出部分集中在main文件里。验证结果如下：

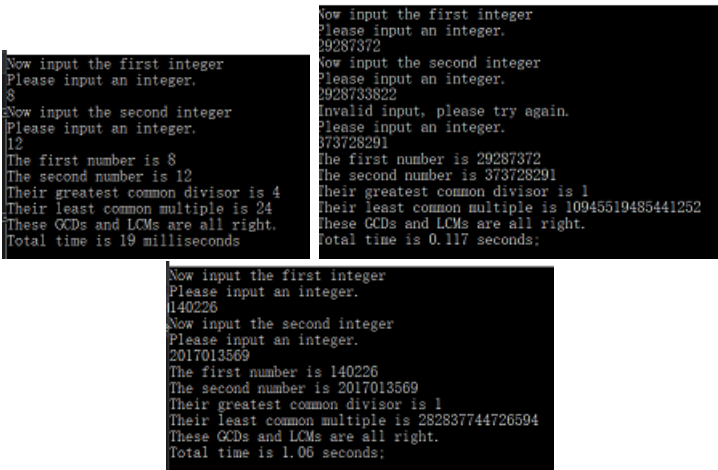


图 4: 测试数据及结果

3 评价

程序对可能的bug进行了规避（包括读入bug，最小公倍数溢出等问题），而且采用辗转相除算法求解问题，提高了运行效率，同时能机器生成随机数自动验证。