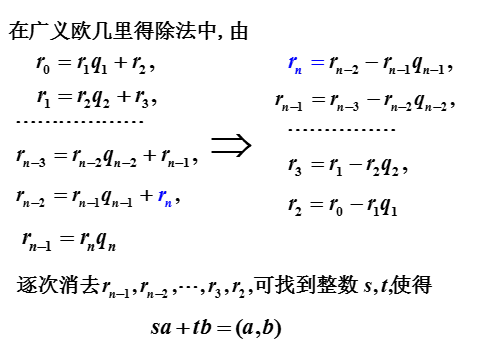
最大公因数的算法

14061075 修闽珂

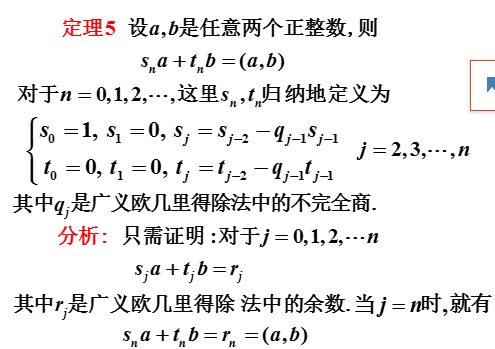
1. 算法原理

最大公因数的算法以欧几里得算法为基础。欧几里得算法的精髓就在于求两个大数的公因数可以转化为求两个比较小的数的最大公因数。即r2 = r0 – q \* r1,则(r0, r1)= (r1, r2)。通过一系列的循环可以将两个比较大的数的最大公因数求出来。这里运用了两种方法来实现这种算法。第一种被称为回代法。



第二种算法被称为扩展欧几里得算法。

这两种方法都是将最大公因数表示为sa + tb = rn=(a, b)的形式。



1. 测试数据

|  |  |
| --- | --- |
| 737 635 | 193 -224 1 |
| 12345678912 23456 | 35 -18421673 32 |
| 1573 286 | 1 -5 143 |
| 4 -2 | 0 -1 2 |
| 46480 39423 | 16720 -19713 1 |
| 0 165 | 0 1 165 |
|  |  |

1. 心得体会

欧几里得算法可以说是求最大公约数的很好的思路。如果不用这种方法而是用我们小学时学的方法，处理很大的两个数步骤就会急剧增加，严重降低效率。欧几里得算法可以很快地将两个大的数“缩小”到合适的范围内。而逐次回代法和扩展欧几里得定理方法有类似之处。

1. 算法流程

输入a, b

如果a或者b是小于零的情况

是

引入一个转换数，用以转化数据的符号并将输入的ab转换为正数

否

将两个数中较大的那个赋值给r0,较小的那个赋值给r1

如果有至少一个数等于零

是

得到结果，跳出循环

否

r2 = r0 % r1

q = int ( r0 / r1 )

是

如果大的数字恰好就是小的数的倍数

输出的情况就是

0，1，r1，结束

是

否

如果大的数并不恰好是小的数的倍数

在一个列表中储存所有的数字，列表的最后一个就是所求的s和t

是否能够整除

否

是

输出加上正负号的结果，结束

输出时间信息

在一个列表中储存所有的数字，列表的最后一个就是所求的s和t

两种算法的 不同。第一种方法不记录每次的商

第二种方法每次的商都要记录。第一种方法中每次循环求的s，t都已经直接利用了本次循环求得的商。第二种方法则是要利用上一次循环求得的商。

两个程序占用内存的情况

K:\Users\Hunter Hugh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获.png

回代法

K:\Users\Hunter Hugh\AppData\Local\Microsoft\Windows\INetCache\Content.Word\捕获1.png

扩展欧几里得方法