gcd方法原理-最大公约数

<https://blog.csdn.net/zd454909951/article/details/59108505>

# “碾转相除法”求最大公约数

设用户输入的两个整数为n1和n2且n1>n2，余数=n1%n2。当余数不为0时，把除数赋给n1做被除数，把余数赋给n2做除数再求得新余数，若还不为0再重复知道余数为0，此时n2就为最大公约数。

例：gcd(20,8)

20=2\*8+4

8=2\*4 因此gcd(20,8)=4

*/\*\*  
 \* 求最大公约数  
 \** ***@param*** *num1  
 \** ***@param*** *num2  
 \** ***@return*** *\*/*public static int gcd(int num1, int num2) {  
 if (num1 < num2) {  
 int temp = num1;  
 num1 = num2;  
 num2 = temp;  
 }  
 int remainder = num1 % num2;  
 while (remainder != 0) {  
 num1 = num2;  
 num2 = remainder;  
 remainder = num1 % num2;  
 }  
 return num2;  
}

# 其他方法：不建议

## 从小到大找

假设输入的两个整数为n1和n2，检查k(k=2,3,4…)是否为n1和n2的最大公约数，直到k大于两个数中较小的一个。

int gcd=1,k=1;

while(k<=num1 && k<=num2)

{

if(num1%k==0 && num2%k==0)

gcd=k;

k++;

}

return gcd;

## 从大到小找

假设输入的两个整数为n1和n2，首先求n1和n2的最小值d，然后依次检验d，d-1，d-2，….，1是否是n1和n2的公约数，这样找到的第一个公约数就是最大公约数。

int d;

if(num1<num2)

d=num1;

else

d=num2;

while(d>=1){

if(num1%d==0 && num2%d==0)

break;

d--;

}

return d;