Euclid's Algorithm

Given three integers a,b,c, can you write c in the form

c=ax+by

for integers x and y? If so, is there more than one solution? Can you find them all? Before answering this, let us answer a seemingly unrelated question:

How do you find the greatest common divisor (gcd) of two integers a,b?

We denote the greatest common divisor of a and b by gcd(a,b), or sometimes even just (a,b). If (a,b)=1 we say a and b are *coprime*.

The obvious answer is to list all the divisors a and b, and look for the greatest one they have in common. However, this requires a and b to be factorized, and no one knows how to do this efficiently.

A few simple observations lead to a far superior method: Euclid’s algorithm, or the Euclidean algorithm. First, if d divides a and d divides b, then d divides their difference, a - b, where a is the larger of the two. But this means we’ve shrunk the original problem: now we just need to find gcd(a,a−b). We repeat until we reach a trivial case.

Hence we can find gcd(a,b) by doing something that most people learn in primary school: division and remainder. We give an example and leave the proof of the general case to the reader.

Suppose we wish to compute gcd(27,33). First, we divide the bigger one by the smaller one:

33=1×27+6

Thus gcd(33,27)=gcd(27,6). Repeating this trick:

27=4×6+3

and we see gcd(27,6)=gcd(6,3). Lastly,

6=2×3+0

Since 6 is a perfect multiple of 3, gcd(6,3)=3, and we have found that gcd(33,27)=3.

This algorithm does not require factorizing numbers, and is fast. We obtain a crude bound for the number of steps required by observing that if we divide a by b to get a=bq+r, and r>b/2, then in the next step we get a remainder r′≤b/2. Thus every two steps, the numbers shrink by at least one bit.

# Euclid's Algorithm

Euclid's Algorithm appears as the solution to the Proposition VII.2 in the Elements:

*Given two numbers not prime to one another, to find their greatest common measure.*

What Euclid called "common measure" is termed nowadays a [common factor*or a*common divisor](https://www.cut-the-knot.org/arithmetic/FactorsAndMultiples.shtml). Euclid VII.2 then offers an [*algorithm*](https://www.cut-the-knot.org/WhatIs/WhatIsAlgorithm.shtml) for finding the [*greatest common divisor*](https://www.cut-the-knot.org/blue/chinese.shtml#gcd) (gcd) of two integers. Not surprisingly, the algorithm bears Euclid's name.

The algorithm is based on the following two observations:

1. If b|a then *[gcd](https://www.cut-the-knot.org/blue/chinese.shtml)*(a, b) = b.

This is indeed so because no number (b, in particular) may have a divisor greater than the number itself (I am talking here of non-negative integers.)

1. If a = bt + r, for integers t and r, then gcd(a, b) = gcd(b, r).

Indeed, every common divisor of a and b also divides r. Thus gcd(a, b) divides r. But, of course, gcd(a, b)|b. Therefore, gcd(a, b) is a common divisor of b and r and hence gcd(a, b) ≤ gcd(b, r). The reverse is also true because every divisor of b and r also divides a.

#### **Example**

Let a = 2322, b = 654.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2322 = 654·3 + 360 |  | gcd(2322, 654) = gcd(654, 360) |
|  | 654 = 360·1 + 294 |  | gcd(654, 360) = gcd(360, 294) |
|  | 360 = 294·1 + 66 |  | gcd(360, 294) = gcd(294, 66) |
|  | 294 = 66·4 + 30 |  | gcd(294, 66) = gcd(66, 30) |
|  | 66 = 30·2 + 6 |  | gcd(66, 30) = gcd(30, 6) |
|  | 30 = 6·5 |  | gcd(30, 6) = 6 |

Therefore, gcd(2322,654) = 6.



For any pair a and b, the algorithm is bound to terminate since every new step generates a similar problem (that of finding gcd) for a pair of smaller integers. Let Eulen(a, b) denote the length of the Euclidean algorithm for a pair a, b. Eulen(2322, 654) = 6, Eulen(30, 6) = 1. I'll use this notation in the proof of the following very important consequence of the algorithm:

## **Corollary**

*For every pair of whole numbers a and b there are two integers s and t such that as + bt = gcd(a, b).*

### **Example**

2322×20 + 654×(-71) = 6.

## **Proof**

Let a > b. The proof is by induction on Eulen(a, b). If Eulen(a, b) = 1, i.e., if b|a, then a = bu for an integer u. Hence, a + (1 - u)b = b = gcd(a, b). We can take s = 1 and t = 1 - u.

Assume the Corollary has been established for all pairs of numbers for which Eulen is less than n. Let Eulen(a, b) = n. Apply one step of the algorithm: a = bu + r. Eulen(b, r) = n - 1. By the inductive assumption, there exist x and y such that bx + ry = gcd(b,r) = gcd(a,b). Express r as r = a - bu. Hence, ry = ay - buy; bx + (ay - buy) = gcd(a, b). Finally, b(x - uy) + ay = gcd(a, b) and we can take s = x - uy and t = y.