Euclid's algorithm

Consider two natural numbers a and b. Usually a is allowed to be an integer (i.e., it can be negative), but to keep things simple here we will say that $a, b \in \mathbb{N}$, a and b are examples of counting or natural numbers, also known as the positive integers.

We can find their *greatest common divisor*, written (a, b). First we write the unique prime factorization of a and b:

```
180 =           2 x 2 x 3 x 3 x 5
140 =           2 x 2 x         5 x 7
gcd(140,180) =  2 x 2 x         5 = 20
```

Pick out the common factors and the $\gcd(a, b)$ will be their product. (We will develop a theorem on unique prime factorization in another chapter).

However, it is important that we do not need to actually factor a and b, as we'll see.

The algorithm works like this. Find integers $r \geq 0$ and $q > 0$ such that

$$a = b \cdot q + r$$

• If $r = 0$ we are done: b divides a equally. Otherwise

∘ switch $a = b$ and $b = r$ and repeat.

Then b is the gcd of the original a and b.

In our example

```
180 = 140 x 1 + 40
140 =  40 x 3 + 20
 40 =  20 x 2 + 0
gcd =  20
```

Proof.

Let $n = a + b$ and suppose that p evenly divides a and b, that is, p is a common factor of both.

Then $a = px$ and $b = py$ so

$$n = a + b = px + py = p(x + y)$$

p evenly divides $a + b$.

More important for us, p

$$a - b = px - py = p(x - y)$$

p evenly divides $a - b$.

□

To speed things up, we find the largest multiple of b, mb such that

$$mb < a < (m + 1)b$$

And then we repeat, finding the common factor of b and $a - mb$.

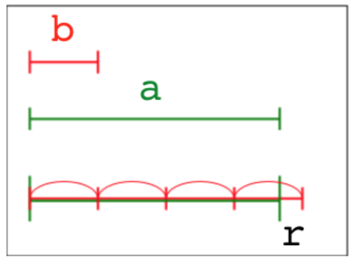
longer proof

Here is the reason this works. First, we can always find q and r such that

$$a = b \cdot q + r$$

where $0 <= r < b$ (since if $r = b$, then $a = b \cdot (q + 1) + 0$.

This is a version of the Archimedean property for positive integers.



It may be paraphrased by saying

> given a bathtub full of water and a teaspoon, it is possible to empty the bathtub.

Either $a = b \cdot q$ and we are done or:

$$b \cdot q < a < b \cdot q + b$$

So then

$$a - bq > 0$$

$$a - bq < b$$

With $r = a - bq$, we obtain $0 < r < b$.

Let u be the largest integer that divides both a and b (the greatest common divisor)

$$a = su$$

$$b = tu$$

Then

$$su = q \cdot tu + r$$

$$r = su - q \cdot tu$$

$$r = u(s - q \cdot t)$$

So u divides r.

Hence every common divisor of a and b is also a divisor of b and r.

recursive program

Here are two examples of programs in different styles that implement the algorithm (with no error checking):

```
def gcd(a,b):
    r = a % b
    if r == 0:
        return b
    return gcd(b,r)
```

```
def gcd(a,b):
    r = a % b
    while r != 0:
        a,b = b,r
        r = a % b
    return b
```

The first version is *recursive*, it may call itself. The second uses a **while** loop to accomplish the same thing.

We're using the built-in mod function % from Python, but could do something like this:

```
def mod(a,b):
    if a == 0 or b == 0:
        raise ValueError
    if a == b:
        return 0
    if a < b:
        a,b = b,a
```

```
c = b + b
if c > a:
    return a - b
next = c + b

while True:
    if next > a:
        return a - c
    c = next
    next = c + b
```