Greatest common divisors: an introduction

0.1. Division with remainder (recall)

If a and b are two integers with $b > 0$, then there is a unique pair (q, r) with

$$q \in \mathbb{Z} \qquad \text{and} \qquad r \in \{0, 1, \ldots, b-1\} \qquad \text{and} \qquad a = qb + r.$$

This pair (q, r) is called the **quo-rem pair** of a and b. Specifically, q is called the **quotient** and r the **remainder** of the division of a by b. We use the notations

$$a//b := q \qquad \text{and} \qquad a\%b := r.$$

For example,

$$\begin{aligned} 7//3 &= 2 \quad \text{and} \quad 7\%3 = 1; \\ 6//3 &= 2 \quad \text{and} \quad 6\%3 = 0; \\ (-1)//3 &= -1 \quad \text{and} \quad (-1)\%3 = 2. \end{aligned}$$

0.2. Greatest common divisors

0.2.1. Definition

Definition 0.2.1. Let a and b be two integers.

(a) The **common divisors** of a and b are the integers that divide both a and b simultaneously.

(b) The **greatest common divisor** of a and b is the largest of all common divisors of a and b, provided that a and b are not both 0. If a and b are both 0, it is defined to be 0 instead.

We denote the greatest common divisor of a and b by $\gcd(a, b)$, and we refer to it as the **gcd** of a and b.

Some examples:

- What is $\gcd(4, 6)$?

  The divisors of 4 are $-4, -2, -1, 1, 2, 4$.

  The divisors of 6 are $-6, -3, -2, -1, 1, 2, 3, 6$.

  Thus, the common divisors of 4 and 6 are $-2, -1, 1, 2$.

  So the greatest common divisor of 4 and 6 is 2. That is, $\gcd(4, 6) = 2$.

- What is $\gcd(0, 5)$?

  The divisors of 0 are all integers (since $0 = x \cdot 0$ for each integer x).

  The divisors of 5 are $-5, -1, 1, 5$.

  So the common divisors of 0 and 5 are $-5, -1, 1, 5$.

  Thus, $\gcd(0, 5) = 5$.

- By definition, $\gcd(0,0) = 0$. It is not literally the largest of all common divisors of 0 and 0, because there is no largest integer. (This is why we had to make an exception for the case "a and b are both 0" when defining the gcd!)

Let us check that $\gcd(a,b)$ is well-defined – i.e., that if a and b are two integers that are not both 0, then there **is** a largest among all common divisors of a and b. In fact:

- The number 1 is always a common divisor of a and b, so there is **at least one** common divisor of a and b.
- At least one of the numbers a and b is nonzero and thus has only finitely many divisors, so there are **only finitely many** common divisors of a and b.

Hence, the set of all common divisors of a and b is nonempty and finite, and thus has a largest element. This shows that $\gcd(a,b)$ is well-defined.

The above definition gives a stupid but functional algorithm to compute $\gcd(a,b)$: Just run over all divisors of a and check which of them divide b (at least if $a \neq 0$). This is slow when a is a large number.

However, is there a better way?

0.2.2. Basic properties

There is one, and in order to find it, we shall first show a sequence of basic properties of gcds. First, some very simple results:

Proposition 0.2.2. Let $a, b \in \mathbb{Z}$. Then, $\gcd(a,b)$ is a nonnegative integer.

Proof. If a and b are both 0, this gcd is 0. Else, it is ≥ 1 (since 1 is a common divisor). □

Proposition 0.2.3. We have $\gcd(a,b) = \gcd(b,a)$ for all $a, b \in \mathbb{Z}$.

Proof. Obvious (a and b play equal roles in the definition). □

Proposition 0.2.4. We have $\gcd(a,b) \mid a$ and $\gcd(a,b) \mid b$.

Proof. Obvious for $a = b = 0$. Also obvious otherwise. □

Proposition 0.2.5. Let $a, b \in \mathbb{Z}$. Then, $\gcd(-a,b) = \gcd(a,b)$ and $\gcd(a,-b) = \gcd(a,b)$.

Proof. For any integer c, the number $-c$ has the same divisors as c. Thus, the common divisors of $-a$ and b are exactly those of a and b. Hence, in particular, $\gcd(-a,b) = \gcd(a,b)$. Similarly, we can show $\gcd(a,-b) = \gcd(a,b)$. □

Proposition 0.2.6. Let $a \in \mathbb{Z}$. Then, $\gcd(a,0) = \gcd(0,a) = |a|$.

Proof. Every integer divides 0. Thus, the common divisors of a and 0 are precisely the divisors of a. If a is nonzero, then the greatest among those is $|a|$, so that Proposition 0.2.6 holds. If $a = 0$, then $\gcd(0,0) = 0$ gets us to the same conclusion. □

0.2.3. The Euclidean recursion

Next we shall show something subtler:

Proposition 0.2.7. Let $a, b, u \in \mathbb{Z}$. Then,

$$\gcd(a,\ ua+b) = \gcd(a,b).$$

In other words, the gcd of two integers does not change if you add a multiple of one to the other.

Proof. This is trivial if $a = 0$ (because $ua + b = b$ in this case), so let us assume that $a \neq 0$. Thus, both gcds are taken in the literal sense (i.e., they are literally greatest among the common divisors of the respective numbers).

We shall show that the common divisors of a and $ua + b$ are precisely the common divisors of a and b. To prove this, we must prove the following two claims:

> *Claim 1:* Every common divisor of a and $ua + b$ is a common divisor of a and b.
>
> *Claim 2:* Every common divisor of a and b is a common divisor of a and $ua + b$.

Proof of Claim 2. Let d be a common divisor of a and b. We must prove that d is a common divisor of a and $ua + b$. In other words, we must prove that $d \mid a$ and $d \mid ua + b$.

Of course, $d \mid a$ is clear by definition of d. Moreover, from $d \mid a \mid ua$ and $d \mid b$, we see that the two numbers ua and b are multiples of d. Hence, their sum $ua + b$ is a multiple of d as well (since a sum of two multiples of d is again a multiple of d). In other words, $d \mid ua + b$. Thus, $d \mid a$ and $d \mid ua + b$ are both proved. In other words, Claim 2 is proved. □

Proof of Claim 1. Let $c = ua + b$. Then, $(-u)\,a + c = \underbrace{(-u)\,a + ua}_{=0} + b = b$. However, Claim 1 (applied to $-u$ and c instead of u and b) yields that every common divisor of a and $(-u)\,a + c$ is a common divisor of a and c. In other words, every common divisor of a and b is a common divisor of a and $ua + b$ (since $(-u)\,a + c = b$ and $c = ua + b$). This proves Claim 1. □

Having proved both claims, we conclude that the common divisors of a and $ua+b$ are precisely the common divisors of a and b. Thus, the greatest common divisor of a and $ua+b$ is the greatest common divisor of a and b. This proves Proposition 0.2.7. □

Corollary 0.2.8 (Euclidean recursion for the gcd)**.** Let $a \in \mathbb{Z}$, and let b be a positive integer. Then,

$$\gcd(a,b) = \gcd(b,\ a\%b).$$

Proof. Let q and r be the quotient and the remainder of the division of a by b. Then, $a = qb + r$ (by the definition of quotient and remainder) and $a\%b = r$ (by the definition of $a\%b$).

Now, Proposition 0.2.3 yields

$$\begin{aligned}
\gcd(a,b) &= \gcd(b,a) = \gcd(b,\ qb+r) && (\text{since } a = qb+r) \\
&= \gcd(b,r) && \left(\begin{array}{c}\text{by Proposition 0.2.7,}\\ \text{applied to } b,r,q \text{ instead of } a,b,c\end{array}\right) \\
&= \gcd(b,\ a\%b) && (\text{since } r = a\%b).
\end{aligned}$$

This proves Corollary 0.2.8. □

0.2.4. The Euclidean algorithm

Let us put this corollary to some use: We shall compute some gcds by repeatedly simplifying them using Corollary 0.2.8. For instance,

$$\begin{aligned}
\gcd(179,\ 18) &= \gcd(18,\ 179\%18) && (\text{by Corollary 0.2.8}) \\
&= \gcd(18,\ 17) && (\text{since } 179\%18 = 17) \\
&= \gcd(17,\ 18\%17) && (\text{by Corollary 0.2.8}) \\
&= \gcd(17,\ 1) && (\text{since } 18\%17 = 1) \\
&= \gcd(1,\ 17\%1) && (\text{by Corollary 0.2.8}) \\
&= \gcd(1,\ 0) && (\text{since } 17\%1 = 0) \\
&= |1| && (\text{by Proposition 0.2.6}) \\
&= 1
\end{aligned}$$

and

$$\begin{aligned}
\gcd(73,\ 333) &= \gcd(333,\ 73\%333) && \text{(by Corollary 0.2.8)} \\
&= \gcd(333,\ 73) \\
&= \gcd(73,\ 333\%73) && \text{(by Corollary 0.2.8)} \\
&= \gcd(73,\ 41) \\
&= \gcd(41,\ 73\%41) && \text{(by Corollary 0.2.8)} \\
&= \gcd(41,\ 32) \\
&= \gcd(32,\ 41\%32) && \text{(by Corollary 0.2.8)} \\
&= \gcd(32,\ 9) \\
&= \gcd(9,\ 32\%9) && \text{(by Corollary 0.2.8)} \\
&= \gcd(9,\ 5) \\
&= \gcd(5,\ 9\%5) && \text{(by Corollary 0.2.8)} \\
&= \gcd(5,\ 4) \\
&= \gcd(4,\ 5\%4) && \text{(by Corollary 0.2.8)} \\
&= \gcd(4,\ 1) \\
&= \gcd(1,\ 4\%1) && \text{(by Corollary 0.2.8)} \\
&= \gcd(1,\ 0) = |1| = 0.
\end{aligned}$$

These two computations are instances of a general algorithm for computing $\gcd(a, b)$ for any two integers a and b, where $b \geq 0$. This algorithm proceeds as follows:

- If $b = 0$, then the gcd is $|a|$.
- If $b > 0$, then we replace a and b by b and $a\%b$ and recurse (i.e., we apply the same algorithm again to b and $a\%b$ instead of a and b).

In Python, this algorithm can be implemented as follows:

```
def gcd(a, b): # for b nonnegative
  if b == 0:
    return abs(a) # this is |a|
  return gcd(b, a%b)
```

This algorithm is called the **Euclidean algorithm**. It will terminate because at each step, b (the second argument) gets smaller:

$$a\%b < b \qquad (\text{since } a\%b \in \{0, 1, \ldots, b-1\} \text{ by the definition of remainders}).$$

Moreover, it will terminate fairly quickly, because of the following fact:

Proposition 0.2.9. Assume that a and b are positive integers. Then, at every step of the Euclidean algorithm except perhaps for the first step, the product ab drops by at least a factor of 2.

Proof sketch. A step replaces a pair (a, b) by $(b,\ a\%b)$. Note that $b > a\%b$ (since $a\%b \in \{0, 1, \ldots, b-1\}$ by the definition of remainders), so that we always have $a > b$ after the first step.

Now, if $a > b$, then the quotient q and the remainder $r = a\%b$ of the division of a by b satisfy $a = \underbrace{q}_{\substack{\geq 1 \\ (\text{since } a>b)}} b + r \geq \underbrace{b}_{>r} + r > r + r = 2r$, so that $r < \dfrac{a}{2}$. In other words, $a\%b < \dfrac{a}{2}$. Thus, the step that replaces a and b by b and $b\%a$ turns the product ab into $b\underbrace{(a\%b)}_{<\frac{a}{2}} < b \cdot \dfrac{a}{2} = \dfrac{ab}{2}$. Hence, the product drops by at least a factor of 2 when we perform this step. □

Hence, the Euclidean algorithm (when applied to two positive integers a and b) terminates after at most $2 + \log_2(ab)$ many steps. For instance, if a and b are 20-digit integers, then $2 + \log_2(ab) < 2 + \log_2\left(10^{20} \cdot 10^{20}\right) \approx 135$, which is a very manageable number of steps.

Thus, the Euclidean algorithm makes gcds easy to compute. (And because of $\gcd(a, b) = \gcd(a, -b)$, we can easily extend it to the case when b is negative, so that it will work for any integers a and b.)

0.2.5. Bezout's theorem, and the extended Euclidean algorithm

We can tweak the Euclidean algorithm to produce more than just the gcd. But what else could we want?

Assume that your country has only two kinds of coins: 3-cent coins and 5-cent coins. You want to pay exactly 1 cent. You can get change (but only in 3-cent and 5-cent coins). Can you do it, and how?

Yes, you can do this by paying two 5-cent coins and getting three 3-cent coins in return. The reason why this works is that

$$2 \cdot 5 + (-3) \cdot 3 = 1.$$

Similarly, can you pay 1 cent with 5-cent coins and 7-cent coins (with change)? Yes:

$$3 \cdot 7 + (-4) \cdot 5 = 1.$$

Can you pay 1 cent with 4-cent coins and 6-cent coins (with change)? No. In fact, any amount you can pay has the form $x \cdot 4 + y \cdot 6$ for $x, y \in \mathbb{Z}$, and thus is even. But 1 is not even.

In general, the only amounts you could possibly pay with a-cent coins and b-cent coins are multiples of $\gcd(a,b)$. But is this the only requirement? Can you pay exactly $\gcd(a,b)$ cents?

Bezout's theorem says "yes" (if you can get change):

Theorem 0.2.10 (Bezout's theorem)**.** Let a and b be two integers. Then, there exist two integers x and y such that

$$\gcd(a,b) = xa + yb.$$

This theorem is not just amusing but actually very useful. We will sketch a proof by strong induction. The idea of the proof is: You're interested in representing the gcd of two numbers a and b as a multiple of a plus a multiple of b. Such a representation will be called a **Bezout pair** for (a,b). We provide an algorithm for finding such a Bezout pair by piggybacking on the Euclidean algorithm for computing $\gcd(a,b)$.

Let's make this more precise:

Definition 0.2.11. Let a and b be two integers. A **Bezout pair** for (a,b) will mean a pair (x,y) of integers such that

$$\gcd(a,b) = xa + yb.$$

So we must prove that every pair (a,b) of integers has a Bezout pair. We observe the following:

Lemma 0.2.12. For any integer a, there is a Bezout pair for $(a,0)$, namely
$\begin{cases} (1,0), & \text{if } a \geq 0; \\ (-1,0), & \text{if } a < 0. \end{cases}$

Proof. Direct verification, since $\gcd(a,0) = |a|$. □

Lemma 0.2.13. Let a and b be two integers. Let (u,v) be a Bezout pair for $(a,-b)$. Then, $(u,-v)$ is a Bezout pair for (a,b).

Proof. Proposition 0.2.5 yields

$$\begin{aligned} \gcd(a,b) &= \gcd(a,-b) \\ &= ua + v(-b) \qquad (\text{since } (u,v) \text{ is a Bezout pair for } (a,-b)) \\ &= ua + (-v)\, b. \end{aligned}$$

But this shows precisely that $(u,-v)$ is a Bezout pair for (a,b). □

Lemma 0.2.14 (Euclidean recursion for Bezout pairs)**.** Let a and b be two integers, where $b > 0$. Let (u, v) be a Bezout pair for $(b,\ a\%b)$. Then, $(v,\ u - v\,(a//b))$ is a Bezout pair for (a, b).

Proof. Since (u, v) is a Bezout pair for $(b,\ a\%b)$, we have

$$\gcd(b,\ a\%b) = ub + v\,(a\%b).$$

Also, the definition of quotient and remainder yields $a = qb + r$, where $q = a//b$ and $r = a\%b$. In other words,

$$a = (a//b)\,b + (a\%b).$$

Hence,

$$a\%b = a - (a//b)\,b.$$

Now, Corollary 0.2.8 yields

$$\begin{aligned}\gcd(a, b) = \gcd(b,\ a\%b) &= ub + v\ \underbrace{(a\%b)}_{=a-(a//b)b} \\ &= ub + v\,(a - (a//b)\,b) \\ &= ub + va - v\,(a//b)\,b \\ &= (u - v\,(a//b))\,b + va \\ &= va + (u - v\,(a//b))\,b.\end{aligned}$$

This shows that $(v,\ u - v\,(a//b))$ is a Bezout pair for (a, b). □

Using Lemma 0.2.12 and Lemma 0.2.14, we can give a recursive algorithm for computing Bezout pairs (here implemented in Python):

```
def bezout_pair(a, b): # for b nonnegative
  if b == 0: # this is the trivial case
    if a >= 0:
      return (1, 0)
    if a < 0:
      return (-1, 0)
  # now to the nontrivial case (b > 0):
  (u, v) = bezout_pair(b, a%b)
  return (v, u - v * (a//b))
```

In human language, this says:

- If $b = 0$, then a Bezout pair for (a, b) is either $(1, 0)$ or $(-1, 0)$ depending on whether $a \geq 0$ or $a < 0$.
- Otherwise, replace a and b by b and $a\%b$, then compute a Bezout pair (u, v) for the new a and b, and then replace it by $(v,\ u - v\,(a//b))$ where a and b are again the original a and b (not the new a and b).

This algorithm is called the **extended Euclidean algorithm**. It can be easily adapted to negative b using Lemma 0.2.13. It is comparably fast to the original Euclidean algorithm, but not quite as fast, since it involves "back-substitution" (the step where (u, v) is transformed into $(v,\ u - v\,(a//b))$).

0.2.6. The universal property of the gcd

Consider two integers a and b that are not both 0. By its definition, $\gcd(a, b)$ is greater or equal to any common divisor of a and b. But something even better is true: It is **divisible** by any common divisor of a and b. In other words:

Proposition 0.2.15. Let a and b be two integers. Then, any common divisor of a and b is a divisor of $\gcd(a, b)$.

Proof. Let d be a common divisor of a and b. We must prove that d is a divisor of $\gcd(a, b)$.

Bezout's theorem yields $\gcd(a, b) = xa + yb$ for some integers x and y. Consider these x and y. Both xa and yb are multiples of d (since $d \mid a \mid xa$ and $d \mid b \mid yb$). Thus, their sum $xa + yb$ is also a multiple of d (since a sum of two multiples of d is a multiple of d). But this sum is $\gcd(a, b)$. So we have shown that $\gcd(a, b)$ is a multiple of d. In other words, d is a divisor of $\gcd(a, b)$. This proves Proposition 0.2.15. □

Theorem 0.2.16 (universal property of the gcd)**.** Let a and b be two integers. Then, the common divisors of a and b are precisely the divisors of $\gcd(a, b)$.

Proof. Proposition 0.2.15 shows that any common divisor of a and b is a divisor of $\gcd(a, b)$. Conversely, any divisor d of $\gcd(a, b)$ is a common divisor of a and b (since $d \mid \gcd(a, b) \mid a$ and likewise $d \mid b$). Theorem 0.2.16 follows. □

0.2.7. The Frobenius coin problem

Let us return to the puzzle about a-cent coins and b-cent coins. What denominations can you pay using these coins if you do **not** take change?

Usually, you cannot pay $\gcd(a, b)$ cents. For example, you cannot pay 2 cents with 4-cent and 6-cent coins. But still, you can pay $4, 6, 8, 10, 12, \ldots$ cents.

With 3-cent coins and 5-cent coins, you can pay the denominations

$$3,\ 5,\ 6,\ \underbrace{8,\ 9,\ 10,\ 11,\ 12,\ 13,\ \ldots}_{\text{any integer } \geq 8}.$$

How does the answer look like for arbitrary a and b ? (This is known as the **Frobenius coin problem**.)

The only interesting case is when $\gcd(a, b) = 1$, because if a and b have common divisors larger than 1, then we can just factor these divisors out.

In the case $\gcd(a, b) = 1$, there is a nice partial answer by Sylvester:

Theorem 0.2.17 (Sylvester's two-coin theorem)**.** Let a and b be two positive integers such that $\gcd(a,b) = 1$. Then:

(a) Any integer $n > ab - a - b$ can be written in the form

$$n = xa + yb \qquad \text{with } x, y \text{ nonnegative integers}$$

(i.e., you can pay n cents with a-cent coins and b-cent coins, without having to take change).

(b) The integer $ab - a - b$ cannot be written in this form.

(c) Among the first $(a-1)(b-1)$ nonnegative integers $0, 1, \ldots, ab - a - b$, exactly half can be written in the form

$$n = xa + yb \qquad \text{with } x, y \text{ nonnegative integers,}$$

while the other half cannot.

(d) For any integer n, exactly one of the two integers n and $ab - a - b - n$ can be written in this form.

Proof. See Theorem 3.8.3 in Lecture 11 of `https://www.cip.ifi.lmu.de/~grinberg/t/23wd` (note: I did not actually do this in class). □