EUCLID'S EXTENDED ALGORITHM AND APPLICATIONS

FRANKLIN

The foundation of the algorithm is the following theorem

Theorem 1: Euclidean division theorem

If a, b are integers, then there exist integers q, r, such that

$$a = bq + r$$
$$0 \leq r < |b|$$

Extended Euclidean Algorithm

Input: Two integers a and b. The input could also be two polynomials, or in general anything that satisfies Theorem 1.

Algorithm:

- **Step 1:** If $b > a$, then swap a and b. In other words, assume that $a \geq b$.
- **Step 2:** Call $r_0 = a$ and $r_1 = b$ and set $i = 0$.
- **Step 3:** While $r_{i+1} \neq 0$, divide r_i by r_{i+1} as in Theorem 1

  $$r_i = r_{i+1} q_i + r_{i+2}$$

  and increment i.

Output: The algorithm outputs all the equations obtained in the loop of Step 2. These are

$$\begin{aligned}
r_0 &= r_1 q_0 + r_2 \\
r_1 &= r_2 q_1 + r_3 \\
r_2 &= r_3 q_2 + r_4 \\
\vdots \; & \;\; \vdots \\
r_{n-1} &= r_n q_{n-1} + r_{n+1} \\
r_n &= r_{n+1} q_n + 0
\end{aligned}$$

The algorithm always ends when the last remainder computed is 0.

Information that can be read from the output of Euclid's algorithm

- The $\gcd(a, b)$ satisfies

  $$\gcd(a, b) = r_{n+1}$$

  This is, the greatest common divisor of a and b is the last number used as divisor when the algorithm terminated.

- Bezout's equation expressing $\gcd(a, b)$ as a combination
$$ax + by = \gcd(a, b)$$
for some integers x and y.

Bezout's equation

To produce Bezout's equation for a, b, from the output of Euclid's extended algorithm we first write the output by solving in each equation for the remainder

$$\begin{aligned}
r_0 - r_1 q_0 &= r_2 \\
r_1 - r_2 q_1 &= r_3 \\
r_2 - r_3 q_2 &= r_4 \\
\vdots \quad & \quad \vdots \\
r_{n-2} - r_{n-1} q_{n-2} + &= r_n \\
r_{n-1} - r_n q_{n-1} + &= r_{n+1} \\
r_n &= r_{n+1} q_n + 0
\end{aligned}$$

Substituting all of these equations into the second to last one, we get $\gcd(a, b)$ expressed in terms of r_0 and r_1, which are a and b, as wanted.

Multiplicative inverse in modular arithmetic

In the case that $\gcd(a, b) = 1$, Bezout's equation takes the form
$$ax + by = 1$$
If we reduce this equation modulo b, the term bx becomes 0, for being a multiple of b. We obtain
$$ax \equiv 1 \pmod{b}$$
Therefore, x, or rather its remainder modulo b, is the multiplicative inverse of a, modulo b.

Numeric example

Let us compute the gcd of 2519 and 377.
First, we perform Euclid's Extended Algorithm.

$$\begin{aligned}
2519 &= 377 \cdot 6 + 257 \\
377 &= 257 \cdot 1 + 120 \\
257 &= 120 \cdot 2 + 17 \\
120 &= 17 \cdot 7 + \boxed{1} \\
17 &= \boxed{1} \cdot 17 + 0
\end{aligned}$$

The last divisor used is the gcd. So, $\gcd(2519, 377) = 1$.

Now, to form Bezout's equation, let's solve for the remainders in all these equations and substitute each into the next one until we get to the second-to-last equation. We don't want to carry out any of the arithmetic operations, while we are doing the substitutions, at least not the ones with the numbers 2519 and 377.

$$2519 - 377 \cdot 6 = 257$$

$$377 - 257 \cdot 1 = 120$$

$$257 - 120 \cdot 2 = 17$$

$$120 - 17 \cdot 7 = \boxed{1}$$

Substituting the third (second-to-last) equation into the last one to eliminate the remainder 17, we get

$$120 - (257 - 120 \cdot 2) \cdot 7 = \boxed{1}$$

Now we can use the second equation to eliminate from this one all occurrences of the remainder 120. We get

$$(377 - 257 \cdot 1) - (257 - (377 - 257 \cdot 1) \cdot 2) \cdot 7 = \boxed{1}$$

Now we use the first equation to eliminate from this one all occurences of the remainder 257. We get

$$(377 - (2519 - 377 \cdot 6) \cdot 1) - ((2519 - 377 \cdot 6) - (377 - (2519 - 377 \cdot 6) \cdot 1) \cdot 2) \cdot 7 = \boxed{1}$$

Finally, we gather together all terms that are multiplied by 2519 and all that are multiplied by 377. We get

$$2519 \cdot (-22) + 377 \cdot (147) = \boxed{1}$$

The two factors -22 and 147 in Bezout's equation are not unique. We could, for example add and subtract a multiple of $2519 \cdot 377$ and get

$$\begin{aligned}\boxed{1} &= 2519 \cdot (-22) + 2519 \cdot 377 \cdot k - 2519 \cdot 377 \cdot k + 377 \cdot (147) \\ &= 2519 \cdot (377 \cdot k - 22) + 377 \cdot (147 - 2519 \cdot k)\end{aligned}$$

So, the factors $377 \cdot k - 22$ and $147 - 2519 \cdot k$ also work.

In the process we also got that

$$377 \cdot 147 \equiv 1 \pmod{2519}$$

and that

$$2519 \cdot (-22) \equiv 1 \pmod{377}$$

Implementation

It is important to practice the computations above a few times, by hand. To verify your computations we can use a computer.

The following is a function in Python that inputs a and b and returns a triple d, x, y such that

$$ax + by = d$$

```
def xgcd(a, b):
    """return (d, x, y) such that a*x + b*y = d = gcd(a, b)"""
    x0, x1, y0, y1 = 0, 1, 1, 0
    while a != 0:
        (q, a), b = divmod(b, a), a
        y0, y1 = y1, y0 - q * y1
        x0, x1 = x1, x0 - q * x1
    return b, x0, y0
```

.

Note: To copy this code into a Python interpreter, remember that in Python the indentation of the lines is important. So, give 4 spaces to indent each indented line.

Executing xgcd(2519,377) we get (1,−22,147). So, the example should be OK (or both the example and the Python code are wrong).

Continued fractions

A byproduct of the output of the Extended Euclid's Algorithm for the input a, b, with $a \geq b$, is a *continued fraction* expansion of the rational number $\frac{a}{b}$. With the notation above, we get that

$$\frac{a}{b} = \frac{r_0}{r_1} = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots + \cfrac{1}{q_n}}}}$$