Tutorial 12

Gidon Rosalki

2025-01-30

1 Algorithms on numbers (RSA)

1.1 Extended Euclidean algorithm

1.1.1 Operations on algorithms

Let there be a numbers $n \in \mathbb{N}$. We may represent it as $\sum_{i=0}^{\lceil \log(n) \rceil} b_i 2^i$.

Let there be $a \geq b > 0 \in \mathbb{N}$, and let us write the length of the representation of a, as k, which is to say: $\log(a) \approx k$.

Addition/Subtraction: $O(k)$
Multiplication/Division: $O(k^2)$
Modulo: $O(k^2)$
Division: Let there be $a, b \in \mathbb{N},\ a \geq b$, we will say that b divides a, and write $a \mid b$ if there exists $c \in \mathbb{N}$ such that $a = c \cdot b$.
Division with remainder: The division with remainder of a by b is $a = c \cdot b + r$, where $c = \left\lfloor \frac{a}{b} \right\rfloor$.
Modulo: $a \mod b = r = a - bc$ where $0 \leq r \leq b - 1$
The greatest common divisor (GCD): of $a, b \in \mathbb{N}$ let there be:

$$gcd(a, b) = \max \{d \in \mathbb{N} : d \mid a \wedge d \mid b\}$$

if $gcd(a, b) = 1$ then a, b are not divisible. Some points:

- $gcd(a, 0) = 0$
- $gcd(0, 0) = NaN$

1.1.2 GCD

Theorem 1 (Lemma)**.** *$gcd(a, b) = gcd(b, a \mod b)$, where $a, b \in \mathbb{N}_0$*

Proof . Let $gcd(a, b) = d,\ gcd(b, a \mod b) = d'$. We want to prove that $d' \leq d \wedge d \leq d'$.
$d \mid d'$: This implies that d divides a, b, therefore:

$$\begin{aligned}
a &\stackrel{def}{=} k_1 \cdot d \\
b &\stackrel{def}{=} k_2 \cdot d \\
b &\stackrel{def}{=} k_3 \cdot d' \\
b &\stackrel{def}{=} k_4 \cdot d'
\end{aligned}$$

Let us start using these.

$$\begin{aligned}
c &= \left\lfloor \frac{a}{b} \right\rfloor \\
\implies a &= c \cdot b + a \mod b \\
&= c \cdot k_3 d' + k_4 d' \\
&= d' (c \cdot k_3 + k_4) \\
&\implies d' \mid a
\end{aligned}$$

Therefore d' is a joint divisor of a, b. d is the largest divisor, and therefore $d \geq d'$.

$$\begin{aligned} a \mod b &= a - c \cdot b \\ &= k_1 \cdot d - c \cdot k_2 d \\ &= d\left(k_1 - c \cdot k_2\right) \\ &\implies d \mid a \mod b \end{aligned}$$

d divides b from the definition, and therefore d is a common divisor of b, and $a \mod b$. d' is the largest divisor, and therefore $d \leq d'$. □

Theorem 2 (Lemma)**.**

$$gcd\left(a, b\right) = \min\left\{z' : z = ax + by, : x, y \in \mathbb{Z}\right\}$$

Proof . We will write $S = \{ax + by \geq 1 | x, y \in \mathbb{Z}\}$, and $z = \min\{S\}$, $t = gcd\left(a, b\right)$.

$t \mid z\left(t \leq z\right)$:

$$\begin{aligned} z &= ax + by \\ &= \left(k_1 t\right) x + \left(k_2 t\right) y \\ &= t\left(k_1 x + k_2 y\right) \\ \implies t \cdot \bar{c} &= z \\ \implies t \mid z &\implies t \leq z \end{aligned}$$

($t \geq 1, z \geq 1$, therefore $k_1 x + k_2 y > 0$, and also $k_1 t + k_2 y \in \mathbb{Z}$, therefore $k_1 x + k_2 y \in \mathbb{N}$)

$z \leq t$: We will suppose that $z \mid a, z \mid b$, and is therefore a common divisor. As we saw earlier, $t \geq z$
$z \leq a$: $a \in S$ since $x = 1, y = 0$ are possible. By definition $z \in S$, and by definition $z = \min\{S\} \leq a$.
Let us divide a by z:

$$a = c \cdot z + r$$

where $r = a \mod z$. We will show that $r = 0$, and get that $z \mid a$ by definition.
Let us assume that $r \neq 0, 1 \leq r \leq z - 1$.

$$\begin{aligned} r &= a - c \cdot z \\ &= a - c \cdot \left(ax + by\right) \\ &= a\left(1 - cx\right) + \left(-y\right) b \\ &= ax_0 + by_0 \\ \implies r &\in S \end{aligned}$$

Since z is minimal, $z \leq r$. On the other hand, $r = a \mod z$ which implies that $z > z$, which is a contradiction. Therefore $r = 0$, which is to say that $a = c \cdot z$, which is to say $z \mid a$.

In order to show the other direction, we may rely on symmetry, and we have proven both directions. □

1.1.3 Extended Euclidean algorithm

Input: $a, b \in \mathbb{N},\ a \geq b,\ a \geq 1$
Output: $g = gcd\left(a, b\right) \in \mathbb{N} \wedge x, y\left(\in \mathbb{Z}\right) : ax + by = g$

Naive algorithm: We will go from 1 until b, and check if the number divides a, and b. This takes b iterations, however $b = 2^{\log(b)}$, so if $k = \log\left(b\right)$, then this takes $O\left(2^k\right)$.

Theorem 3. *The algorithm is correct, which is to say returns* (g, x, y) *such that* $g = ax + by = gcd\left(a, b\right)$

Proof . Induction on the number of recursive calls $= R$.
Base: No recursive calls: $R = 0$, so there are no calls. This occurs when $b = 0$, so from the definition, $gcd\left(a, 0\right) = a$, and $a = 1 \cdot a + 0 \cdot 0 = a$.

EE 1

Input: *input*

Output: *output*

if $b = 0$ **then**

  return $(a, 1, 0)$

else

  $(g', x', y') = EE\left(b, a \mod b\right)$

end if

return $\left(g', y', x' - y'\left\lfloor \frac{a}{b} \right\rfloor\right)$

Inductive step: Let us assume that the call $R-1$ returns the correct output, and we will prove for R.
From the assumption, g, x', y' enable that

$$\begin{aligned} g' &= gcd\left(b, a \mod b\right) \\ &= x' \cdot b' + \left(a \mod b\right) y' \end{aligned}$$

From lemma 1

$$g' = gcd\left(a, b\right)$$

and therefore returning g' is correct. We want to show that

$$g' = y' \cdot a + \left(x' - y'\left\lfloor \frac{a}{b} \right\rfloor\right) b$$

$$a \mod b = a - \left\lfloor \frac{a}{b} \right\rfloor b$$

$$\begin{aligned} g' &= x'b + y'\left(a - \left\lfloor \frac{a}{b} \right\rfloor b\right) \\ &= y'a + \left(x' - \left\lfloor \frac{a}{b} \right\rfloor y'\right) b \end{aligned}$$

and so the recursive call returns a correct output. □

Runtime: A single iteration takes $O\left(k^2\right)$ where $k = \log\left(a\right)$. We will show that $a \mod b \leq \frac{a}{2}$. We will get that every 2 iterations, the numbers are smaller by a factor of 2, and thus the maximum number of recursive calls is 2 times the number of times that we need to divide b by 2 to get to 0.
Let us prove that $a \mod b \leq \frac{a}{2}$.

- $b \leq \frac{a}{2}$: $0 \leq a \mod b \leq b - 1 < \frac{a}{2}$
- $b > \frac{a}{2}$: $a \mod b = a - b < \frac{a}{2}$

The operation of dividing by two is simply *shiftright*, and we may perform it at most the length of the binary representation, and so $2\log\left(b\right)$ is the maximum possible number of recursive calls. Therefore $O\left(k \cdot k^2\right) = O\left(k^3\right)$

2 RSA

Rivest-Shamir-Adleman. We are looking for two very large prime numbers p, q. Let $n = pq$. We will find e has no common divisor with $LCM\left(\left(p-1\right), \left(q-1\right)\right)$. We will find $de = 1 \mod \left(p-1\right)\left(q-1\right)$, and publish d, n.

$$\begin{aligned} f\left(x\right) &= x^d \mod n = y \\ f^{-1}\left(y\right) &= y^e \mod n = x \end{aligned}$$

How does this relate to EE? Finding d.

$$\begin{gathered} EE\left(\left(p-1\right)\left(q-1\right), e\right) \\ xe + y\left(p-1\right)\left(q-1\right) = 1 \\ xe = 1 \mod \left(p-1\right)\left(q-1\right) \end{gathered}$$

So $x = d$, and we have found it.