4a. Number theory: integer division and prime numbers

Grzegorz Kosiorowski

Krakow University of Economics

1. Divisibility, GCD, Euclidean Algorithms
2. Lowest common multiple
3. Prime numbers

Number theory - introduction

Number theory is a branch of mathematics devoted to studying properties of numbers, mostly integers (in this chapter, any number is an integer unless stated otherwise). It was once perceived as the most abstract branch of mathematics without any direct applications. Today, in the age of computers and information, we know this view is no longer applicable.

Number theory - applications

- Modern systems of encryption and decryption are based on the results of number theory, such as the difficulty of factoring large integers: we are going to focus on this application in the next part of the course.
- Effective methods of data compression and (pseudo)random numbers generation.
- Signal processing and data analysis through the so-called Fast Fourier Transform.
- Efficient error-correcting codes (such as: numbers of bank accounts and credit cards, digital signals).
- Studying musical scales and generating music (Brian Eno).
- Applications to modern theoretical physics: http://empslocal.ex.ac.uk/people/staff/mrwatkin/zeta/physics.htm

Computational complexity for basic operations

- Encryption of a number n is $O(\log n)$.
- Complexity of addition of numbers a and b is $O(\log a + \log b)$.
- Complexity of multiplication of numbers a and b is $O(\log a \log b)$.

Division on integers

Addition, subtraction and multiplication on the set of integers have no particular properties that need to be studied: the rules are the same as in $\mathbb{R}$. However, the division of integers is more interesting as the result of dividing two integers in $\mathbb{R}$ does not have to be an integer. Thus, the integer division must be defined differently.

Quotient and remainder

Let a, b be integers and let q, r be such that $a = bq + r$ and $0 \leqslant r < b$. Then, q is termed a *quotient* of a and b, and r is termed a *remainder* of the division of a by b. We denote $r = a \mod b$.

$$17 = 5 \cdot 3 + 2$$

$$2 = 17 \mod 5$$

Divisibility

Divisibility

b divides a (or *a is divisible* by *b*, or *b* is a *divisor* of *a*, or *a* is a *multiple* of *b*) if there exists q such that $a = bq$, or, equivalently, if $a \bmod b = 0$. We denote $b|a$.

Example: $3|6$.

Divisibility properties

Divisibility properties

For any $a, b, c \neq 0$ it holds that:
a) if $a|b$, then $a|bc$,
b) if $a|b$ and $b|c$, then $a|c$,
c) if $a|b$ and $a|c$, then $a|(b+c)$.

Greatest common divisor

Greatest common divisor

Greatest common divisor of two non-zero a and b is the largest number d such that $d|a$ and $d|b$. We denote it by $GCD(a, b)$

Properties of GCD

For non-zero numbers a, b:
$GCD(a, 1) = 1$, $GCD(a, a) = a$, $GCD(a, b) = GCD(b, a)$.

Examples: $GCD(9, 6) = 3$, $GCD(36, 12) = 12$, $GCD(33, 26) = 1$,
For larger numbers we need an algorithmic approach.

Euclidean Algorithm

The algorithm below is already about 2300 years old. It appeared first in *Elements* by Euclid. Its aim is to find the GCD of two integers.

Euclidean Algorithm

Input: Positive integers a, b.
Variables: r (integer).

- I. While $b \neq 0$ execute:
- Ia. $r := a \mod b$.
- Ib. $a := b$, $b := r$.
- **Output**: The value of a at the end of the algorithm is the GCD of the input values.

Run time of the Euclidean algorithm for $a \geqslant b$ is $O(\log^3 a)$.

Euclidean Algorithm - example

Example

Find $GCD(888, 1104)$.

First we define the input $a = 888$, $b = 1104$.
We are going to present the subsequent steps of the algorithm in the following table:

Step	r	a	b
1			
2			
...			

Euclidean Algorithm - example

Example

Find $GCD(888, 1104)$.

$b \neq 0$, thus we perform the first step: $r = a \mod b = 888$ (because $a = 0 \cdot 1104 + 888$)

Next, we substitute $a := b = 1104$, and $b := r = 888$. The first step only changed the ordering of a and b (thus, it is simpler to just establish the input for the Euclidean Algorithm such that $a > b$)

Step	r	a	b
1	888	1104	888
2			

Euclidean Algorithm - example

Example

Find $GCD(888, 1104)$.

$b \neq 0$, so $r = 1104 \mod 888 = 216$ (because $1104 = 1 \cdot 888 + 216$)
Next, $a := b = 888$, and $b := r = 216$.

Step	r	a	b
1	888	1104	888
2	216	888	216

Euclidean Algorithm - example

Example

Find $GCD(888, 1104)$.

Still $b \neq 0$, so $r = 888 \mod 216 = 24$ (because $888 = 4 \cdot 216 + 24$)
Next, $a := b = 216$, and $b := r = 24$.

Step	r	a	b
1	888	1104	888
2	216	888	216
3	24	216	24

Euclidean Algorithm - example

Example

Find $GCD(888, 1104)$.

Still $b \neq 0$, so $r = 216 \mod 24 = 0$ (because $216 = 9 \cdot 24$)
Next, $a := b = 24$, and $b := r = 0$. $b = 0$, thus the algorithm ends and $GCD(888, 1104) = a = 24$.

Step	r	a	b
1	888	1104	888
2	216	888	216
3	24	216	24
4	0	24	0

Extended Euclidean Algorithm and Bezout's Identity

A generalization of the Euclidian algorithm, the so-called extended Euclidean algorithm, is particularly useful for cryptology. This algorithm aims to find Bezout coefficients of two integers, namely, the numbers x and y from the following theorem:

Bezout's Identity

For any positive integers a and b there exist integers x and y such that

$$ax + by = GCD(a, b).$$

Extended Euclidean Algorithm

Extended Euclidean Algorithm - part I

Input: Positive integers a, b.

Variables: r_i, q_i, x_i,y_i - sequences of integers, i - loop counter.

- I. $r_0 := a$, $r_1 := b$,$i := 1$.
- II. While $r_i \neq 0$ execute:
- IIa. $i := i + 1$.
- IIb. $r_i := r_{i-2} \mod r_{i-1}$, $q_{i-1} := (r_{i-2} - r_i)/r_{i-1}$.
- When this loop ends, we obtain $r_{i-1} = \mathrm{GCD}(a, b)$ and for all $k < i - 1$ we have $r_k - q_{k+1} r_{k+1} = r_{k+2}$.

Extended Euclidean Algorithm

Extended Euclidean Algorithm - part II

- III. $i := i - 1$, $x_i := 0$, $y_i := 1$.
- IV. While $i > 1$ execute:
- IVa. $i := i - 1$.
- IVb. $x_i := y_{i+1}$, $y_i := x_{i+1} - q_i x_i$.
- **Output**: (x_1, y_1) is a pair of Bezout coefficients for the input of the algorithm. GCD was obtained in part I.

Extended Euclidean Algorithm - example

Example

Find GCD$(234, 123)$ and integers x, y such that $234x + 123y = GCD(234, 123)$.

To begin with, we substitute $a = 234$, $b = 123$.
We are going to present the subsequent steps of the algorithm in the following table:

i	r_i	q_i	x_i	y_i
0	?			
1	?	?	?	?
2	?	?	?	?
...	...	...	...	...

Extended Euclidean Algorithm - example

Example

Find GCD$(234, 123)$ and integers x, y such that $234x + 123y = GCD(234, 123)$.

For stage I, we insert a and b into their places in the table and set $i = 1$:

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

Extended Euclidean Algorithm - example

i	r_i	q_i	x_i	y_i
0	234			
1	123	?	?	?
2	?	?	?	?
...	...	...	...	...

$r_i = r_1 \neq 0$, so we continue the first loop: $i = 2$, $234 = 1 \cdot 123 + 111$, thus $r_2 = 111$ and $q_1 = 1$.

Extended Euclidean Algorithm - example

$r_i = r_1 \neq 0$, so we continue the first loop: $i = 2$, $234 = 1 \cdot 123 + 111$, thus $r_2 = 111$ and $q_1 = 1$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

Extended Euclidean Algorithm - example

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	?	?	?
3	?	?	?	?
...	...	...	...	...

$r_i = r_2 \neq 0$, so we continue: $i = 3$, $123 = 1 \cdot 111 + 12$, thus $r_3 = 12$ and $q_2 = 1$.

Extended Euclidean Algorithm - example

$r_i = r_2 \neq 0$, so we continue: $i = 3$, $123 = 1 \cdot 111 + 12$, thus $r_3 = 12$ and $q_2 = 1$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

Extended Euclidean Algorithm - example

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	?	?	?
4	?	?	?	?

$r_i = r_3 \neq 0$, so we continue: $i = 4$, $111 = 9 \cdot 12 + 3$, thus $r_4 = 3$ and $q_3 = 9$.

Extended Euclidean Algorithm - example

$r_i = r_3 \neq 0$, so we continue $i = 4$, $111 = 9 \cdot 12 + 3$, thus $r_4 = 3$ and $q_3 = 9$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

Extended Euclidean Algorithm - example

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	?	?	?

$r_i = r_4 \neq 0$, so we continue: $i = 5$, $12 = 4 \cdot 3 + 0$, thus $r_5 = 0$ and $q_3 = 4$.

Extended Euclidean Algorithm - example

... thus $r_5 = 0$ and $q_3 = 4$. r_i=0 so the first loop (stage II of the algorithm) ends. We already have $GCD(234, 123) = r_{i-1} = r_4 = 3$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	?	?
5	0	-	-	-

Extended Euclidean Algorithm - example

For stage III we move the loop counter one step back: $i = 4$ and we set $x_i = x_4 = 0$ and $y_i = y_4 = 1$. From now on, we move "up the table".

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	0	1
5	0	-	-	-

Extended Euclidean Algorithm - example

$x_i := y_{i+1}$, $y_i := x_{i+1} - q_i x_i$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	?	?
4	3	4	0	1
5	0	-	-	-

$i > 1$, so we reduce the loop counter by 1 ($i = 3$) and calculate:

$x_3 := y_4 = 1$; $y_3 := x_4 - q_3 x_3 = 0 - 9 \cdot 1 = -9$.

Extended Euclidean Algorithm - example

...$(i = 3)$ and calculate: $x_3 := y_4 = 1$;
$y_3 := x_4 - q_3 x_3 = 0 - 9 \cdot 1 = -9$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Extended Euclidean Algorithm - example

$x_i := y_{i+1}$, $y_i := x_{i+1} - q_i x_i$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	?	?
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, so we reduce the loop counter by 1 ($i = 2$) and calculate:

$x_2 := y_3 = -9$; $y_2 := x_3 - q_2 x_2 = 1 - 1 \cdot (-9) = 10$.

Extended Euclidean Algorithm - example

...$(i = 2)$ and calculate: $x_2 := y_3 = -9$;
$y_2 := x_3 - q_2 x_2 = 1 - 1 \cdot (-9) = 10$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Extended Euclidean Algorithm - example

$x_i := y_{i+1}$, $y_i := x_{i+1} - q_i x_i$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	?	?
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

$i > 1$, so we reduce the loop counter by 1 ($i = 1$) and calculate:

$x_1 := y_2 = 10$; $y_1 := x_2 - q_1 x_1 = -9 - 1 \cdot 10 = -19$.

Extended Euclidean Algorithm - example

...$(i = 1)$ and calculate: $x_1 := y_2 = 10$;
$y_1 := x_2 - q_1 x_1 = -9 - 1 \cdot 10 = -19$.

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	10	-19
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Extended Euclidean Algorithm - example

i	r_i	q_i	x_i	y_i
0	234			
1	123	1	10	-19
2	111	1	-9	10
3	12	9	1	-9
4	3	4	0	1
5	0	-	-	-

Now $i = 1$, so the stage III and the entire algorithm ends. From the table we obtain the output: $GCD(234, 123) = 3$ and $234 \cdot 10 + 123 \cdot (-19) = 3$.

Lowest common multiple

Lowest common multiple

Lowest common multiple of integers a and b (denoted LCM(a, b)) is the smallest positive integer w such that $a|w$ i $b|w$.

Properties of LCM

For positive integers a, b:
$LCM(a, 1) = a$, $LCM(a, a) = a$, $LCM(a, b) = LCM(b, a)$.

Examples: $LCM(7, 8) = 56$, $LCM(9, 6) = 18$.
Again, we need an algorithmic approach for larger numbers.

Lowest common multiple - algorithmic approach

Relation of GCD and LCM

For positive integers a, b it holds that:

$$a \cdot b = GCD(a, b) \cdot LCM(a, b).$$

In particular: $GCD(a, b) = \frac{ab}{LCM(a,b)}$ and $LCM(a, b) = \frac{ab}{GCD(a,b)}$.

We can use the theorem above to find LCM algorithmically: one needs to perform the Euclidean Algorithm to find $GCD(a, b)$ and then apply the formula $LCM(a, b) = \frac{ab}{GCD(a,b)}$.

Lowest common multiple and divisibility by multiple numbers

The following theorem will be useful for combinatorics (counting elements of sets):

Divisibility by two numbers

For positive integers a, b it holds that:

$$a|c \text{ and } b|c \Leftrightarrow LCM(a,b)|c.$$

Naturally, the theorem can be easily generalized into the case of more than two numbers.

Prime numbers - introduction

Each number $a > 1$ has **at least** 2 divisors: 1 and itself. However, some of them stand out:

Prime number

A *prime number* (or just a *prime*) is a natural number that has exactly two different divisors: 1 and a. Any natural number greater than 1 that is not a prime is termed a *composite* number.

0 and 1 are consider neither prime nor composite numbers. The set of all primes is denoted by $\mathcal{P}$.

Coprime numbers

If $\mathrm{GCD}(a, b) = 1$, then a and b are *coprime* (or *relatively prime*) to each other. We denote this relation as $a \perp b$.

The Euclidean algorithm determines if two numbers are coprime.

Prime numbers: relevance

Prime numbers are crucial for number theory because each integer can be factorized into a product of prime numbers in exactly one way (if we ignore changes in their order). Primes are thus considered "building blocks" of all numbers and compared to atoms as "building blocks" of molecules (obviously, this comparison is a bit exaggerated but it illustates well the significance of primes).

Prime factorization of a positive integer

A prime factorization of a positive integer is a representation of this number as a product of prime powers namely: $n = p_1^{a_1} p_2^{a_2} \cdot \ldots \cdot p_k^{a_k}$, where $p_1, \ldots p_k$ are different primes and $a_1, \ldots, a_k$ are positive integers.

$$600 = 2^3 \cdot 3 \cdot 5^2 = 5^2 \cdot 2^3 \cdot 3 \ldots$$

Fundamental theorem of arithmetics

Fundamental theorem of arithmetics

Any integer $n > 1$ can be represented uniquely (up to the order of the factors) by its prime factorization.

While the theorem itself is important, its significance for computer science lies in the fact that this theorem is not constructive, namely, no efficient way of finding a prime factorization for a given number is known. This observation is a key factor for the validity of modern cryptographic systems and other "ratchet algorithms". Naturally, small numbers are easy to factorize. The most difficult are these numbers, which are products of two large primes.

Time complexity of factorization

- At the moment, there exists no publicly known algorithm applicable to non-quantum hardware that factorizes any natural number in polynomial time. However, there is also no proof that such an algorithm does not exist.
- For now (2025) the best asymptotic result for an algorithm factoring any n is

$$O\left(e^{\left(\left(\frac{8}{3}\right)^{\frac{2}{3}}+o(1)\right)(\log n)^{\frac{1}{3}}(\log\log n)^{\frac{2}{3}}}\right)$$

  achieved by the algorithm GNFS in 1993. There are many algorithms working in "subexpotential time", namely better than $O((1+\varepsilon)^n)$ for any $\varepsilon > 0$.
- The Shor's algorithm factorizes natural numbers in time $O(n^3)$, but it requires a quantum computer to work.

Primality tests

- Although the efficient factorization of integers is for now impossible, testing if a natural number is prime is much easier (important for "ratchet algorithms"!)
- The AKS algorithm (Agrawal, Kayal, Saxen, 2002) is theoritically the fastest deterministic primality test with a run time of $O(\log^6 n)$, where n is a tested number. However, it is impractical in real-life applications (is very slow for "reasonably large" numbers).
- In practice, probabilistic primality tests are most commonly used. They do not guarantee that they result is always correct, but either they are always correct for "reasonably large numbers" or the confidence percentage is sufficiently high to be practically deterministic). In the same time, they are much faster. Examples: Fermat, Miller-Rabin, Sollovay-Strassen, Baille-PSW tests).

Infinitude of primes

There is not "largest" prime due to the following theorem (also proven by Euclid).

Infinitude of primes

There are infinitely many primes.

The original proof is short and beautiful, thus this is the only proof of this course required for the exam.

Proof of infinitude of primes

Infinitude of primes

There are infinitely many primes.

Proof by contradiction: Assume that $p_1, \ldots, p_k$ are all the primes (so there are k of them). Let $n = p_1 p_2 \ldots p_k + 1$. n is either prime or composite. n cannot be prime, because it is larger than any of numbers $p_1, \ldots, p_k$ and, by our initial assumption, they are all the primes. If n is composite, it has a unique prime factorization thus it is divisible by at least one prime number. However, for any $i \in \{1, 2, \ldots, k\}$ the integer division of n by p_i is $n = p_i \cdot q + 1$, where $q = p_1 p_2 \ldots p_{i-1} \cdot p_{i+1} \ldots p_k$ (a product of all p but p_i) so the remainder of this division is 1. Thus, n is not divisible by any prime so it cannot be composite. Therefore, the assumption that there are finitely many primes leads to a false conclusion, thus the assumption itself must be false. QED.

The largest known prime number

Pure curiosity (numbers that large are not practically applicable even for cryptology): the largest **known** (namely, such that its primality is proven) prime number (for 1 I 2025) is $2^{136279841} - 1$. It has 41024320 digits in the decimal system. It was found in October 2024 by Luke Durant from GIMPS (Great Internet Mersenne Prime Search).