3.2. The Euclidean Algorithm

3.2.1. The Division Algorithm. The following result is known as The Division Algorithm:\footnote{The result is not really an “algorithm”, it is just a mathematical theorem. There are, however, algorithms that allow us to compute the quotient and the remainder in an integer division.} If \( a, b \in \mathbb{Z}, \ b > 0, \) then there exist unique \( q, r \in \mathbb{Z} \) such that \( a = qb + r, \ 0 \leq r < b. \) Here \( q \) is called quotient of the integer division of \( a \) by \( b, \) and \( r \) is called remainder.

3.2.2. Divisibility. Given two integers \( a, b, b \neq 0, \) we say that \( b \) divides \( a, \) written \( b \mid a, \) if there is some integer \( q \) such that \( a = bq: \)

\[
b \mid a \iff \exists q, \ a = bq.
\]

We also say that \( b \) divides or is a divisor of \( a, \) or that \( a \) is a multiple of \( b. \)

3.2.3. Prime Numbers. A prime number is an integer \( p \geq 2 \) whose only positive divisors are 1 and \( p. \) Any integer \( n \geq 2 \) that is not prime is called composite. A non-trivial divisor of \( n \geq 2 \) is a divisor \( d \) of \( n \) such that \( 1 < d < n, \) so \( n \geq 2 \) is composite iff it has non-trivial divisors. Warning: 1 is not considered either prime or composite.

Some results about prime numbers:

1. For all \( n \geq 2 \) there is some prime \( p \) such that \( p \mid n. \)
2. (Euclid) There are infinitely many prime numbers.
3. If \( p \mid ab \) then \( p \mid a \) or \( p \mid b. \) More generally, if \( p \mid a_1 a_2 \ldots a_n \) then \( p \mid a_k \) for some \( k = 1, 2, \ldots, n. \)

3.2.4. The Fundamental Theorem of Arithmetic. Every integer \( n \geq 2 \) can be written as a product of primes uniquely, up to the order of the primes.

It is customary to write the factorization in the following way:

\[
n = p_1^{s_1} p_2^{s_2} \ldots p_k^{s_k},
\]

where all the exponents are positive and the primes are written so that \( p_1 < p_2 < \cdots < p_k. \) For instance:

\[
13104 = 2^4 \cdot 3^2 \cdot 7 \cdot 13.
\]
3.2.5. Greatest Common Divisor. A positive integer $d$ is called a common divisor of the integers $a$ and $b$, if $d$ divides $a$ and $b$. The greatest possible such $d$ is called the greatest common divisor of $a$ and $b$, denoted gcd$(a, b)$. If gcd$(a, b) = 1$ then $a, b$ are called relatively prime.

Example: The set of positive divisors of 12 and 30 is \{1, 2, 3, 6\}. The greatest common divisor of 12 and 30 is gcd(12, 30) = 6.

A few properties of divisors are the following. Let $m, n, d$ be integers. Then:

1. If $d|m$ and $d|n$ then $d|(m + n)$.
2. If $d|m$ and $d|n$ then $d|(m − n)$.
3. If $d|m$ then $d|mn$.

Another important result is the following: Given integers $a, b, c$, the equation

$$ax + by = c$$

has integer solutions if and only if gcd$(a, b)$ divides $c$. That is an example of a Diophantine equation. In general a Diophantine equation is an equation whose solutions must be integers.

Example: We have gcd(12, 30) = 6, and in fact we can write 6 = 1·30 − 2·12. The solution is not unique, for instance 6 = 3·30 − 7·12.

3.2.6. Finding the gcd by Prime Factorization. We have that gcd$(a, b) =$ product of the primes that occur in the prime factorizations of both $a$ and $b$, raised to their lowest exponent. For instance 1440 = $2^5 \cdot 3^2 \cdot 5$, 1512 = $2^3 \cdot 3^3 \cdot 7$, hence gcd(1440, 1512) = $2^3 \cdot 3^2 = 72$.

Factoring numbers is not always a simple task, so finding the gcd by prime factorization might not be a most convenient way to do it, but there are other ways.

3.2.7. The Euclidean Algorithm. Now we examine an alternative method to compute the gcd of two given positive integers $a, b$. The method provides at the same time a solution to the Diophantine equation:

$$ax + by = \text{gcd}(a, b).$$

It is based on the following fact: given two integers $a \geq 0$ and $b > 0$, and $r = a \mod b$, then gcd$(a, b) = \text{gcd}(b, r)$. Proof: Divide $a$ by
b obtaining a quotient q and a remainder r, then

\[ a = bq + r, \quad 0 \leq r < b. \]

If d is a common divisor of a and b then it must be a divisor of \( r = a - bq \). Conversely, if d is a common divisor of b and r then it must divide \( a = bq + r \). So the set of common divisors of a and b and the set of common divisors of b and r are equal, and the greatest common divisor will be the same.

The Euclidean algorithm is as follows. First we divide a by b, obtaining a quotient q and a remainder r. Then we divide b by r, obtaining a new quotient \( q' \) and a remainder \( r' \). Next we divide r by \( r' \), which gives a quotient \( q'' \) and another remainder \( r''' \). We continue dividing each reminder by the next one until obtaining a zero reminder, and which point we stop. The last non-zero remainder is the gcd.

**Example:** Assume that we wish to compute \( \gcd(500, 222) \). Then we arrange the computations in the following way:

\[
\begin{align*}
500 & = 2 \cdot 222 + 56 \quad \rightarrow \quad r = 56 \\
222 & = 3 \cdot 56 + 54 \quad \rightarrow \quad r' = 54 \\
56 & = 1 \cdot 54 + 2 \quad \rightarrow \quad r'' = 2 \\
54 & = 27 \cdot 2 + 0 \quad \rightarrow \quad r''' = 0
\end{align*}
\]

The last nonzero remainder is \( r''' = 2 \), hence \( \gcd(500, 222) = 2 \). Furthermore, if we want to express 2 as a linear combination of 500 and 222, we can do it by working backward:

\[
2 = 56 - 1 \cdot 54 = 56 - 1 \cdot (222 - 3 \cdot 56) = 4 \cdot 56 - 1 \cdot 222
\]

\[= 4 \cdot (500 - 2 \cdot 222) - 1 \cdot 222 = 4 \cdot 500 - 9 \cdot 222.\]

The algorithm to compute the gcd can be written as follows:

```plaintext
1: procedure gcd(a, b)  
2: if a < b then \(/ / \text{make a the largest} \)  
3: \quad swap(a, b)  
4: \quad while b \neq 0 do  
5: \quad \quad begin  
6: \quad \quad \quad r := a \mod b  
7: \quad \quad \quad a := b  
8: \quad \quad \quad b := r  
9: \quad \quad end  
10: \quad return(a)  
11: end gcd
```
The next one is a recursive version of the Euclidean algorithm:

1: procedure gcd_recur(a,b)
2:   if b=0 then
3:      return(a)
4:   else
5:      return(gcd_recur(b,a mod b))
6: end gcd_recur