3. Greatest Common Divisors and the Euclidean Algorithm

Let \(a\) and \(b\) be integers, at least one of them nonzero. An integer \(c\) is called a *common divisor* of \(a\) and \(b\) if \(c \mid a\) and \(c \mid b\). Notice that 1 is a common divisor of every pair of integers. The *greatest common divisor* of \(a\) and \(b\) is (obviously) the largest integer among the set of all common divisors of \(a\) and \(b\). (Since any nonzero integer has only finitely many divisors, there is indeed a largest integer among all the common divisors of \(a\) and \(b\).) We denote the greatest common divisor of \(a\) and \(b\) by \(\gcd(a, b)\). Since 1 is a common divisor of \(a\) and \(b\), \(\gcd(a, b) \geq 1\).

When \(\gcd(a, b) = 1\), we say \(a\) and \(b\) are *relatively prime*.

For small integers, it is often easy to find the \(\gcd\) by “inspection,” i.e., by quickly examining (usually, in our heads) the common divisors. So, for example, it is easy to see that \(\gcd(12, 20) = 4\) and \(\gcd(-22, 15) = 1\). A brute force method for calculating the \(\gcd\) would be to simply find all divisors of the two integers and then find the largest common one. This turns out to be computationally difficult for large integers. (For the same reason, finding the \(\gcd\) using the prime factorizations of the two integers is also difficult, unless we are first given the factorizations.) It turns out, there is a very efficient way of computing the \(\gcd\) of two integers using what is known as the *Euclidean Algorithm*. We first need a basic lemma:

**Lemma 1.** Let \(a, b, q, r\) be integers such that \(a = bq + r\) and \(b \neq 0\). Then \(\gcd(a, b) = \gcd(b, r)\).

**Proof.** Let \(d = \gcd(a, b)\) and \(e = \gcd(b, r)\). Since \(d \mid a\) and \(d \mid b\), then \(d\) divides \(a - bq = r\). So \(d\) is a common divisor of \(b\) and \(r\), hence \(d \leq e\). Similarly, as \(e \mid b\) and \(e \mid r\) then \(e\) divides \(bq + r = a\). Thus, \(e\) is a common divisor of both \(a\) and \(b\), so \(e \leq d\). This proves that \(d = e\).  

Let’s now describe the Euclidean Algorithm:

**The Euclidean Algorithm:** Let \(a\) and \(b\) be two integers (assume \(b \neq 0\)). Dividing \(b\) into \(a\) using the Division Theorem, we get

\[a = bq_1 + r_1\]

with \(0 \leq r_1 < b\). If \(r_1\) is not zero, we can divide \(r_1\) into \(b\):

\[b = r_1q_2 + r_2\]

with \(0 \leq r_2 < r_1\). If \(r_2 \neq 0\), we repeat the process:

\[r_1 = r_2q_3 + r_3\]

with \(0 \leq r_3 < r_2\). Since \(r_1 > r_2 > r_3 > \cdots\) and these integers are all nonnegative, eventually we must reach a remainder of zero: Say

\[r_{n-2} = r_{n-1}q_n + r_n \quad \text{(where } r_n \neq 0\text{), and}\]

\[r_{n-1} = r_nq_{n+1} + 0.\]

Then \(r_n = \gcd(a, b)\).

The proof of this is simply a repeated application of the above lemma:
\[
gcd(a, b) = gcd(b, r_1) = gcd(r_1, r_2) = gcd(r_2, r_3) = \cdots = gcd(r_n, 0) = r_n.
\]

Let’s try this out on an example:

**Example:** Find gcd(141, 120):

\[
\begin{align*}
141 &= 120(1) + 21 \\
120 &= 21(5) + 15 \\
21 &= 15(1) + 6 \\
15 &= 6(2) + 3 \\
6 &= 3(2) + 0
\end{align*}
\]

Thus, \(3 = \gcd(141, 120)\).

Now you try the algorithm on this one:

**Exercise:** Find gcd(1721, 378).

**Homework:**

1. Use the Euclidean Algorithm to find gcd(878, 421).

2. Let \(\{a_n\}\) be the Fibonacci sequence. Prove that gcd\(a_n, a_{n+1}\) = 1 for all \(n \geq 1\).