Euclid’s Theorem:
Let $a,b,q,r$ be positive integers such that $a = qb + r$ then

$$\text{gcd} (a,b) = \text{gcd}(b,r)$$

Euclid’s Algorithm in the Python programming language.

```python
def euclid_gcd(a,b):
    # Assume $a \geq b > 0$
    r = a % b # this returns $r$ such that $a = bq + r$
    while r > 0:
        a,b = b,r
        r = a % b # this returns $r$ s.t. $a = bq + r$
    return b
```
NOTE: The % (mod) operator is found in many programming languages and returns the remainder when doing integer division.

We will argue that euclid_gcd(a,b) finds gcd(a,b) assuming that \( a \geq b > 0 \).

We first argue that the loop terminates, that is \( r \) eventually becomes 0. This is easy to see because the remainder when we divide \( a \) by \( b \) is less than \( b \). The value of \( r \) begins positive and always decreases so it eventually must be zero.

The correctness follows from Euclid’s theorem.

It can also be shown that this function is extremely efficient when compared to looking at all the divisors of \( a \) and \( b \).
Let $a = 250$, and $b = 575$. We can construct a prime factorization of $a$ and $b$.

Prime factorization:
$250 = (2)(5^3)$
$575 = (5^2)(23)$

We can inspect the prime factorization of $a$ and $b$ to obtain a greatest common divisor.

Observe that $5^2$ is the greatest number that divides both $a$ and $b$, that is the $\text{gcd}(a,b)$. Using the prime factorizations of $a$ and $b$ is much less efficient than Euclid’s algorithm. Nevertheless, the prime factorization is useful for obtaining other properties of the greatest common divisor.
Least Common Multiple

Given two non-zero integers $a, b$ we can have many values that are positive common multiples of both $a$ & $b$. By the well ordering principle we know that amongst all of those multiples there is one that is smallest, and this is known as the least common multiple of $a$ and $b$. We can define a function $\text{lcm}(a,b)$ that returns this value.

Example: Suppose $a = 12$, and $b = 24$, so we have $\text{lcm}(a,b) = 24$. In general if $a \mid b$ then $\text{lcm}(a,b) = |b|$. At this point it is worth mentioning that if $a \mid b$ then $\text{gcd}(a,b) = |a|$, and that $\text{lcm}(a,b) \times \text{gcd}(a,b) = |ab|$.

Example: Suppose $a = 13$, and $b = 24$, we have $\text{lcm}(a,b) = (13)(24)$. In general if $a$ and $b$ are relatively prime, that is, if $\text{gcd}(a,b) = 1$ then $\text{lcm}(a,b) = |ab|$

So when $\text{gcd}(a,b) = 1$, we can observe that $\text{lcm}(a,b) \times \text{gcd}(a,b) = |ab|$.
Let a = 250, and b = 575. We can construct a prime factorization of a and b

Prime factorization

\[ 250 = (2)(5^3) \]
\[ 575 = (5^2)(23) \]

We can inspect the prime factorization of a and b to obtain the least common multiple.

\[ 250 \times 575 = (2)(5^3) \times (5^2)(23) = (5^2) \times (2)(5^3)(23) \]

And since gcd(a,b) = 5^2 we can conclude that lcm(a,b) = (2) (5^3) (23).

So in this case we also have:

\[ \text{lcm}(a,b) \times \gcd(a,b) = |ab| \]
Let $p_1, p_2, \ldots, p_k$ denote all of the prime factors of both $a$ and $b$ ordered from smallest to largest. In our example the list of prime factors would be 2, 5, 23.

Let $a_i$ denote the exponent of prime factor $p_i$, for $i, 1 \leq i \leq k$, in a prime factorization of $a$.

In our example $a_1 = 1, a_2 = 3, a_3 = 0$. Similarly we define $b_i$ for $i, 1 \leq i \leq k$. In our example $b_1 = 0, b_2 = 2, b_3 = 1$.

Again referring to our example we have:

\[
\gcd(a,b) = 2^{\min(1,0)} \times 5^{\min(3,2)} \times 23^{\min(0,1)}
\]

and,

\[
\text{lcm}(a,b) = 2^{\max(1,0)} \times 5^{\max(3,2)} \times 23^{\max(0,1)}.
\]

In general using $p_i, a_i$, and $b_i$ as defined above we can express this formula as

\[
\gcd(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times \cdots \times p_k^{\min(a_k, b_k)}
\]

and

\[
\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} \times p_2^{\max(a_2, b_2)} \times \cdots \times p_k^{\max(a_k, b_k)}
\]
One more example

\[ 630 = (2)(3^{2})(5)(7) \]
\[ 84 = (2^{2})(3)(7) \]

By inspection we can see that
\[ \gcd(630, 84) = (2)(3)(7) = 42 \]
\[ \text{And } \text{lcm}(630, 84) = (2^{2})(3^{2})(5)(7) = 1260 \]
Again we have

\[ 630 \times 84 = (2)(3^{2})(5)(7) \times (2^{2})(3)(7) \]
\[ = (2)(3)(7) \times (2^{2})(3^{2})(5)(7) \]
\[ = \gcd(630, 84) \times \text{lcm}(630, 84) \]

These ideas lead to the following theorem that is given without proof.

**Theorem:** Let \( a, b \) be non-zero integers, then

\[ \gcd(a, b) \times \text{lcm}(a, b) = |ab|. \]
Factoring vs. GCD

Factoring an integer \( N \) into its prime factors will use roughly \( \sqrt{N} \) operations.

Computing \( \text{gcd}(N, m) \) with Euclid’s algorithm for \( N > m \geq 0 \) will use roughly \( \log_2 N \) operations.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \log_2 N )</th>
<th>( \sqrt{N} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>10</td>
<td>32</td>
</tr>
<tr>
<td>1099511627776</td>
<td>40</td>
<td>1,048,576</td>
</tr>
<tr>
<td>( 1 \times 10^{301} )</td>
<td>1000</td>
<td>( 3.27 \times 10^{150} )</td>
</tr>
</tbody>
</table>

The efficiency of Euclid’s \( \text{gcd} \) algorithm is essential for implementing current public key crypto systems that are commonly used for e-commerce applications.

With a “key” decoding an encrypted message using Euclid’s algorithm takes about 1000 operations. Without a “key” breaking an encrypted message uses approximately \( 3.27 \times 10^{150} \) operations. This amounts to a small fraction of a second for decoding and many millions of years for breaking the encrypted message.
Congruence Relations

Let $a$ and $b$ be integers. We say that $a$ is congruent to $b$ modulo $m$ written as:

$$a \equiv b \pmod{m}$$

and defined as follows:

$$a \equiv b \pmod{m} \text{ if and only if } m \mid (a-b).$$
Arithmetic with congruences

Suppose we have \( a \equiv b \pmod{m} \) and \( c \equiv d \pmod{m} \).

Then

\[
a + c \equiv (b + d) \pmod{m},
\]

\[
a - c \equiv (b - d) \pmod{m}, \text{ and}
\]

\[
ac \equiv (bd) \pmod{m}.
\]

Examples

\( 5 \equiv 2 \pmod{3} \) and \( 10 \equiv 1 \pmod{3} \)

\[
5 + 10 \equiv (2 + 1) \pmod{3}, \text{ that is, } 15 \equiv 3 \pmod{3}
\]

\[
5 - 10 \equiv (2 - 1) \pmod{3}, \text{ that is, } -5 \equiv 1 \pmod{3}
\]

(Note: By the Division Algorithm Theorem we have \(-5 = (-2)(3) + 1\))

\[
(5)(10) \equiv (2)(1) \pmod{3}, \text{ that is, } 50 \equiv 2 \pmod{3}
\]

These properties require a proof.
Suppose we have $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. Then $a + c \equiv (b + d) \pmod{m}$.

**Proof:** (We need to show that $a + c \equiv (b + d) \pmod{m}$.)

If $a \equiv b \pmod{m}$ then $m \mid (a-b)$. And if $c \equiv d \pmod{m}$ we have $m \mid (c-d)$.

This in turn implies that

$$m \mid ((a - b) + (c - d))$$

which can be written as

$$m \mid ((a + c) - (b + d)).$$

So we can conclude that $a + c \equiv (b + d) \pmod{m}$. □
Suppose we have \( a \equiv b \pmod{m} \) and \( c \equiv d \pmod{m} \). Then \( ac \equiv (bd) \pmod{m} \). □

**Proof:** (We need to show that \( m \mid (ac - bd) \).)

If \( a \equiv b \pmod{m} \) then \( m \mid (a-b) \).
And if \( c \equiv d \pmod{m} \) we have \( m \mid (c-d) \).

This in turn implies that
\[ m \mid (a - b)c \quad \text{because } m \mid (a - b)p \text{ for all integers } p \]
and that
\[ m \mid (c - d)b \quad \text{because } m \mid (a - b)p \text{ for all integers } p. \]

Therefore we have
\[ m \mid ((a - b)c + (c - d)b) \]
Which can be written as:
\[ m \mid (ac - bd) \]

So we can conclude that \( ac \equiv (bd) \pmod{m} \). □
Congruence modulo m is an equivalence relation. Observe that we can partition the integers by their congruences.

**Examples:**

Congruence (mod 2) partitions integers into those that are even and odd.

Congruence (mod 3) partitions integers into three classes those that are divisible by 3 (remainder 0) and those with remainder 1, and remainder 2 when divided by 3.

In general we say that congruence modulo m partitions the integers into m classes called *residue classes modulo* m. Furthermore, each of these residue classes can be denoted by an integer x within the class using the notation \([x]_m\). Using set notation we can express this as follows:

\[
[x]_m = \{a \in \mathbb{Z} : a \equiv x \mod m\}
\]

And each of the residue classes can be denoted by its smallest member as follows:

\([0]_m, [1]_m, [2]_m, ..., [m-1]_m\)
Techniques of Counting (Chapter 5 of SN)

We have already seen and solved several counting problems. For example:

• How many subsets are there of a set with n elements?
• How many two element subsets are there of a set with n elements.
• How many different ways can the numbers in 6-49 draw be chosen?

Counting problems are useful to determine resources used by an algorithm (e.g. time and space).
**Product Rule Principle**

Let $A \times B$ denote the cross product of sets $A$ and $B$. Then $|A \times B| = |A| \times |B|^{1}$

For example suppose you have to pick a main course from: Fish, Beef, Chicken, Vegan. We can write this as the set $M$ (Main), as follows

$$M = \{F, B, C, V\}$$

Furthermore there is also choice of a dessert from: Apple pie, Lemon meringue pie, Ice cream. This can be represented as the set $D$.

$$D = \{A, L, I\}$$

We use the product rule to determine the total number of possible meals, that is:

The **product rule principle** can be stated formally as:

Suppose there is an event $M$ that occurs in $m$ ways and an event $D$ that occurs in $n$ ways, and these two events are *independent* of each other. Then there are $m \times n$ ways for the combination of the two events to occur.

Note that an event can be considered as a set of outcomes, and the combination of events as a cross product of sets.
Product Rule Principle
The rule generalizes to any number of independent sets (events). For example with 3 sets:
Let $A \times B \times C$ denote the cross product of sets $A$, $B$, & $C$.

Then $|A \times B \times C| = |A| \times |B| \times |C|$.

For $k$ sets we have:

$$|A_1 \times A_2 \times \ldots \times A_k| = |A_1| \cdot |A_2| \cdot \ldots \cdot |A_k|$$
For example, DNA is represented using the 4 symbols:

A C G T.

The number of different strings of length 7 using these symbols is:

\[ 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 = 4^7. \]

The number of strings of length \( k \) using these 4 symbols is:

\[ 4^k \]
Sum Rule Principle

Suppose we have the same mains and deserts as before, but must choose a main or a dessert but not both.

Then we have:

$$|\{F,B,C,V\} \cup \{A,L,I\}| = 4 + 3 = 7$$

choices.

Note that these sets have an empty intersection. For non-empty intersections we would need to use the principle of inclusion and exclusion.
The **sum rule principle** can be formally stated as:

Suppose event M can occur in m ways and a second event D can occur in n ways. The number of ways that M or D can occur is m + n.

Again considering an event as a set of outcomes the sum rule principle can be viewed as counting the size of the union of disjoint sets.
Suppose you can take 1 elective from a list of elective courses, where there are 3 courses from the History department, 4 courses from the English department and 2 course from the Psychology department. This can be formalized as the sets:

$$H = \{h_1, h_2, h_3\}, \ E = \{e_1, e_2, e_3, e_4\}, \ P = \{p_1, p_2\}$$

The total number of choices is:

$$|H \cup E \cup P| = |H| + |E| + |P| = 3 + 4 + 2 = 9.$$
The Pigeon Hole Principle

If there are $n$ pigeons, that all must sleep in a pigeon hole, and $n-1$ pigeon holes, then there is at least one pigeon hole where (at least) 2 pigeons sleep.
This should be obvious! Mathematicians give it a name because it is a useful counting tool.

Can we find two people living in the G.T.A. that have exactly the same number of strands of hair on their heads?

The answer is YES! And we can prove it using the pigeon hole principle.
The population of the G.T.A is more than 6 million. (6,418 million 2016 census) Science tells us that nobody has more that 500,000 strands of hair on their heads.

To solve the problem using the pigeon hole principle we imagine 500,000 pigeon holes labelled from 1, ..., 500,000 and then imagine each resident of the G.T.A. entering the pigeon hole labelled with the number of strands of hair on their head. Since 6 million is greater than 500,000 we deduce that there will be at least one pigeon hole where two or more people have entered.
Can we find 13 people living in the G.T.A. that have exactly the same number of strands of hair on their heads?

Again the answer is yes! Can you argue why?

Can we find 2 pairs of people living in the G.T.A. that have exactly the same number of strands of hair on their heads?

The pigeon hole principle is useless for solving this problem and we leave this as an unsolved mystery.
Applications of the Pigeon Hole Principle.

Each of the following problems can be solved using the Pigeon Hole Principle.

1. Show that if 5 points are placed in a 2” by 2” square, then two of them are at most $\sqrt{2}$” apart.

2. Show that among $n+1$ arbitrarily chosen distinct integers, there must be 2 whose difference is divisible by $n$.

3. Prove that in any list of 4 natural numbers $a_1$, $a_2$, $a_3$, $a_4$, there is a string of consecutive numbers whose sum is divisible by 4. (Hint: Consider the 4 values $a_1$, $a_1+a_2$, $a_1+a_2+a_3$, $a_1+a_2+a_3+a_4$)