# Extended Euclidean algorithm implementation

Marlon Mendes

## 1 The Euclidean algorithm

Let $gcd(a, b)$ be the greatest divisor of $a$ and $b$, $a \geq b : a, b \in \mathrm{Z}$, the Euclidean algorithm states that $gcd(a, b) = gcd(b, a \ \% \ b)$, where % denotes the remainder operator. We know from Bézout's Lemma that $gcd(a, b)$ is the smallest positive integer of the form $gcd(a, b) = ax + by : x, y \in \mathrm{Z}$. The question is: how to find $x, y$?

## 2 A recursive solution

We start with the trivial problem: $gcd(a, 0) = a$. What if $a \geq b > 0$? Let $g = gcd(a, b)$. We know that $a = bq + r,\ 0 <= r < |b|$, and $g = gcd(b, a\%b) = gcd(b, r)$.

Suppose we knew how to find $x_1, y_1 : g = bx_1 + ry_1$, can we use that information to build a solution for the original problem $g = ax + by$? Yes.

$$a = bq + r$$
$$g = bx_1 + ry_1$$
$$g = bx_1 + (a - bq)y_1$$
$$g = bx_1 + ay_1 - bqy_1$$
$$g = b(x_1 - qy_1) + a(y_1)$$
$$g = a(y_1) + b(x_1 - qy_1)$$

That last equation builds a solution for the original problem, $g = ax + by$. It says that $x = y_1$ and $y = (x_1 - qy_1)$, where $q = \left\lfloor \frac{a}{b} \right\rfloor$. $x, y$ gives the smallest positive linear combination of $a, b$, but there are others. So we recursively solve the problem of $gcd(b, r)$, and then solve the original problem. Note that the base case of the recursion is $gcd(x, 0) = x$, since the remainder $r$ will always be smaller than $|b|$, the algorithm will eventually ($O(log\ log\ n)$) reach the base case.

# 3 C++ implementation

```cpp
typedef long long ll;

ll ext_gcd(const ll &a, const ll &b, ll &x, ll &y) {
    if(b == 0) {
        x = 1;
        y = 0;
        return a;
    }
    ll x1, y1;
    ll g = ext_gcd(b, a % b, x1, y1);

    const ll q = a / b;
    x = y1;
    y = x1 - y1 * q;
    return g;
}
```

Listing 1: C++ code for the extended gcd

# 4 References


- github.com/edsomjr/TEP/blob/master/Matematica/text/Divisibilidade.md
- Elementary Number Theory, David M. Burton.