# Advanced Euclidean algorithm

While the "normal" Euclidean algorithm simply finds the greatest common divisor of two numbers $a$and $b$, extended Euclidean algorithm finds the GCD also factors in addition to $x$, and $y$such that:

$$a \cdot x + b \cdot y = \gcd(a, b).$$

Ie he finds the coefficients by which the GCD of two numbers expressed in terms of the numbers themselves.

## Algorithm

Make the calculation of these coefficients in the Euclidean algorithm is simple enough to derive formulas by which they change during the transition from couple $(a, b)$to couple $(b\%a, a)$(percent sign we denote modulo arithmetic).

Thus, suppose we have found a solution $(x_1, y_1)$of the problem for the new couple $(b\%a, a)$:

$$(b\%a) \cdot x_1 + a \cdot y_1 = g,$$

and want to get a solution $(x, y)$for our couples $(a, b)$:

$$a \cdot x + b \cdot y = g.$$

To do this, transform the quantity $b\%a$:

$$b\%a = b - \left\lfloor \frac{b}{a} \right\rfloor \cdot a.$$

Substituting this into the above expression $x_1$and $y_1$obtain:

$$g = (b\%a) \cdot x_1 + a \cdot y_1 = \left(b - \left\lfloor \frac{b}{a} \right\rfloor \cdot a\right) \cdot x_1 + a \cdot y_1,$$

and performing regrouping terms, we obtain:

$$g = b \cdot x_1 + a \cdot \left(y_1 - \left\lfloor \frac{b}{a} \right\rfloor \cdot x_1\right).$$

Comparing this with the original expression of the unknown $x$, and $y$we obtain the desired expression:

$$\begin{cases} x = y_1 - \left\lfloor \frac{b}{a} \right\rfloor \cdot x_1, \\ y = x_1. \end{cases}$$

# Implementation

```
int gcd (int a, int b, int & x, int & y) {
	if (a == 0) {
		x = 0; y = 1;
		return b;
	}
	int x1, y1;
	int d = gcd (b%a, a, x1, y1);
	x = y1 - (b / a) * x1;
	y = x1;
	return d;
}
```

This is a recursive function, which still returns the GCD of the numbers $a$and $b$, but beyond that - as desired coefficients $x$and $y$parameters in the form functions are passed by reference.

Recursion base - case $a = 0$. Then the GCD is $b$, and obviously required coefficients $x$and $y$are $0$and $1$, respectively. In other cases, the usual solution is working, and the coefficients are converted by the above formulas.

Advanced Euclidean algorithm in this implementation works correctly even for negative numbers.

# Literature


- Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein. **Algorithms: Design and analysis of** [2005]