

Euclid's Algorithm

Dragan Marković

Euclid's algorithm, is an algorithms which finds greatest common divisor(*GCD*) of two numbers. *GCD*(*a*, *b*) for integers *a* and *b* is such number *d* that *d* divides both *a* and *b*, and there isn't a greater number than *d* with the same property.

lemma. *GCD*(*a*, *b*) = *GCD*(*b*, *a*%*b*), where % is denotes modulo, that is *a*%*b* = *a* - $\lfloor \frac{a}{b} \rfloor \cdot b$

proof. Let *GCD*(*a*, *b*) = *p* and *GCD*(*b*, *a*%*b*) = *q*, then *a* = *b* · *k* + *a*%*b*, since *p*|*a* and *p*|*b* it follows that *p*|(*a*%*b*), likewise sine *q*|*b* and *q*|(*a*%*b*) it follows that *q*|*a*. We have that:

1. *q*|*a* and *q*|*b*
2. *GCD*(*a*, *b*) = *p*

From the definition of greatest common divisor and 1.) and 2.) it follows that *q* ≤ *p*, if *q* > *p* then *GCD*(*a*, *b*) = *q*, which would be a contradiction. Analogical to that:

1. *p*|*b* and *p*|(*a*%*b*)
2. *GCD*(*b*, *a*%*b*) = *q*

We have that *p* ≤ *q*. Since *p* ≤ *q* and *q* ≤ *p*, it must be that *p* = *q*.

We can use this fact to come up with a recursive algorithm for finding greatest common divisor. Simply *GCD*(*a*, *b*) = *GCD*(*b*, *a*%*b*), with the terminating condition that *GCD*(*a*, 0) = *a*. Although it is not at all trivial to prove this algorithm runs in *O*(log *N*) time (worst possible input for this algorithm would be two successive Fibonacci numbers).

Least common multiple (*LCM*), which represents the lowest possible number that is *divisible* by both *a* and *b*, is simply $\frac{a \cdot b}{GCD(a, b)}$.

A couple of examples:

1. *GCD*(5, 2) = *GCD*(2, 1) = *GCD*(1, 0) = 1
2. *GCD*(15, 6) = *GCD*(6, 3) = *GCD*(3, 0) = 3
3. *LCM*(15, 6) = $\frac{15 \cdot 6}{GCD(15, 6)} = \frac{90}{3} = 30$