

Licenciatura en ciencia de la computación



# ALGORITMO EUCLIDEANO

## Matemática Computacional

**Profesor:**  
Nicolas Thériault

**Autor:**  
Sergio Salinas  
Danilo Abellá

## Introducción

### 1 Algoritmo Implementado

A continuación se mostrará que tan eficientes fueron los distintos tipos de algoritmos.

#### 1.1 Algoritmo Euclidiano Simple

Este algoritmo demostró ser altamente ineficiente dado que cuando se trabaja con números muy grandes el tiempo requerido es muy alto, a pesar de que su lógica sea muy simple. Tiempos de espera para los siguientes gcd:

gcd(291,252) - 0.0002230 segundos  
gcd(16261,85652) - 0.0002450 segundos  
gcd(897279761,914407221) - 0.0009650 segundos  
gcd(16534528044,8332745927) - 0.0008580 segundos  
gcd(43263441545690516,43312793054108111) - 0.0068580 segundos

#### 1.2 Algoritmo Euclidiano

Este algoritmo resultó ser la mejor opción en cuanto a complejidad y eficiencia ya que no demostró una lógica muy complicada ni un tiempo de espera demasiado alto, aunque no halla sido de los que menos tiempo de ejecución requirió.

Tiempos de espera para los siguientes gcd:

gcd(291,252) - 0.0000610 segundos  
gcd(16261,85652) - 0.0000780 segundos  
gcd(897279761,914407221) - 0.0000780 segundos  
gcd(16534528044,8332745927) - 0.0000920 segundos  
gcd(43263441545690516,43312793054108111) - 0.0000960 segundos  
gcd(23356764234689876532233456788876543234567,  
6277101735386680763835789423207666416083908700390324961279) - 0.0001140 segundos  
gcd(2123010620889223608977186369097185643295866022  
04480027488784019561937182491149755503041950,  
68185362149486650485123987197513787717187652289362  
47143232679847194284772046122091468887714) - 0.0002170 segundos

### 1.3 Algoritmo Euclidiano Binario

Este sin duda resultó ser el algoritmo con tiempo de espera más corto haciendolo el más eficiente de todos los evaluados anteriormente.

Tiempos de espera para los siguientes gcd:

$\text{gcd}(291, 252)$  - 0.0000510 segundos

$\text{gcd}(16261, 85652)$  - 0.0000580 segundos

$\text{gcd}(897279761, 914407221)$  - 0.0000630 segundos

$\text{gcd}(16534528044, 8332745927)$  - 0.0000610 segundos

$\text{gcd}(43263441545690516, 43312793054108111)$  - 0.0000780 segundos

$\text{gcd}(23356764234689876532233456788876543234567,$

$6277101735386680763835789423207666416083908700390324961279)$  - 0.0000730 segundos

$\text{gcd}(2123010620889223608977186369097185643295866022$

$04480027488784019561937182491149755503041950,$

$68185362149486650485123987197513787717187652289362$

$47143232679847194284772046122091468887714)$  - 0.0001170 segundos

## 2 Formulación experimentos

## 3 Información de Hardware y Software

### 3.1 Notebook - Danilo Abellá

#### 3.1.1 Software

- SO: Xubuntu 16.04.1 LTS
- GMP Library
- Mousepad 0.4.0

#### 3.1.2 Hardware

- AMD Turion(tm) X2 Dual-Core Mobile RM-72 2.10GHz
- Memoria (RAM): 4,00 GB(3,75 GB utilizable)
- Adaptador de pantalla: ATI Raedon HD 3200 Graphics

### 3.2 Notebook - Sergio Salinas

#### 3.2.1 Software

- SO: ubuntu Gnome 16.04 LTS
- Compilador: gcc version 5.4.0 20160609
- Editor de text: Atom

#### 3.2.2 Hardware

- Procesador: Intel Core i7-6500U CPU 2.50GHz x 4
- Video: Intel HD Graphics 520 (Skylake GT2)

## 4 Curvas de desempeño de resultados

## 5 Conclusiones