Algoritmo de Euclides

Encontrar el mínimo común múltiplo de dos enteros

Solución 1:

```
read A, B
mx = max(A,B)
mn = min(A,B)
for M = mx to mx*mn step mx:
     if M % mn == 0:
          print M
          break
```

Número de operaciones:

$$2 + 1 + 1 + min(A, B)(1 + 1 + 1)$$

$$\rightarrow f(N) = 4 + 3\min(A, B)$$

$$\rightarrow O(\min(A, B))$$

Algoritmo de Euclides

No nos sirve para encontrar el mínimo común múltiplo, sino el máximo común divisor de dos enteros

Se basa en que al dividir *M* entre *N*, ambos números enteros, se obtiene un cociente *Q* y un residuo R. Es posible demostrar que el máximo común divisor de *M* y *N* es igual que el de *N* y *R*.

```
read A, B
M, N = A, B
while N ≠ 0:
     M, N = N, M%N
MaxCD = M
```

Ejemplo: *A* es 23, *B* es 13

M = 23, N =13
Iteración 1: M = 13, N = 10
Iteración 2: M = 10, N = 3
Iteración 3: M = 3, N = 1
Iteración 4: M = 1, N = 0
MinCD = 1

Algoritmo de Euclides

Ahora, Si *A* y *B* son enteros, se cumple que:

*A***B* = MinCM(*A*,*B*)*MaxCD(*A*,*B*)

```
read A, B
A, B = M, N
while N ≠ 0:
     M, N = N, M%N
MinCM= A*B/M
```

Ejemplo: *A* es 23, *B* es 13

M = 23, N =13
Iteración 1: M = 13, N = 10
Iteración 2: M = 10, N = 3
Iteración 3: M = 3, N = 1
Iteración 4: M = 1, N = 0
MinCD = 23*13/1 = 299

Algoritmo de Euclides

La complejidad del algoritmo es *O(log(max(A,B))*

La demostración se puede encontrar en:
Stark, H. (1978). An Introduction to Number Theory. MIT Press.

*En Python existen los métodos gcd() y lcm() en la librería math