# ALGORITMOS Y COMPLEJIDAD

# ACTIVIDAD 2

Giampietri Gonzalo - Tomas Perotti

LU:108644 - LU: 105495

Algoritmo analizado: **Algoritmo de Euclides - Maximo Comun Divisor**

**Introducción**

El algoritmo de Euclides, es utilizado para encontrar el máximo común divisor entre dos números enteros. Para esto, mientras el número menor sea mayor a 0 (**m**), se realizará el módulo de **n** por **m** y, se asignará el resultado, como un nuevo valor de **m**. Por otro lado, **n** tomará como nuevo valor el valor previo de **m** y el algoritmo se ejecutará nuevamente.

```
function Euclid(m, n)
    while m > 0 do
        t ← m
        m ← n mod m
        n ← t
    return n
```

**Análisis de performance en tiempo y espacio**

En el caso del Algoritmo de Euclides, a simple vista no podemos determinar una notación asintótica. Si bien el algoritmo solo comprende un “loop”, el mismo no se ejecuta un número fijo de veces, sino que va a depender de los respectivos números de enteros con los que estemos ejecutando el algoritmo.

El total del tiempo tomado por el algoritmo para computar este problema va a estar íntimamente relacionado con la cantidad de veces que el ciclo sea visitado. Por lo tanto, es necesario realizar una serie de cálculos adicionales con el objetivo de obtener una cota superior e inferior relevante a este algoritmo.

A continuación, se procederá a realizar un análisis asintótico del algoritmo:

Al analizar la cantidad de veces que el bucle es recorrido, podemos identificar que dicha cantidad puede ser expresada como un algoritmo recursivo, donde el tiempo de ejecución de este es la cantidad máxima de veces que el bucle es recorrido con sus respectivas entradas. Si tomamos n y m como sus entradas, donde $m \le n \le k$.

Luego, debemos analizar una serie de particularidades, por ejemplo:

Si $n \le 2$, se recorre cero veces (si m = 0) o sólo una vez (si n mod m es 0).

Si ocurre que n mod m es distinto de 0, entonces el valor de n es reducido a la mitad como mínimo y entonces se convierte en k ÷ 2. Por lo tanto, el bucle no es visitado más de T(k ÷ 2) iteraciones.

Con lo cual, podemos asumir la siguiente recurrencia:

La onda es que aca decis que $T(k) \in \Theta(lg(k))$ y abajos revelamos la carta trampa de los números decimales y terminamos diciendo que $T(k) \in \Theta(d)$

$$T(k) \leq \begin{cases} 1 & \text{si } k \leq 2 \\ 2 + T(k \div 2) & \text{caso contrario} \end{cases}$$

Luego, por medio del teorema maestro, si sabemos que $2 \in \Theta(k^{log_2 1})$ entonces $T(k) \in \Theta(k^{log_2 1} lg(k))$ que es igual a:

$$T(k) \in \Theta(lg(k))$$

Por lo tanto, podemos concluir que el algoritmo es del orden $\Theta(lg(k))$.

Por otro lado, si queremos expresar el resultado en función de la longitud de los datos de entrada, al ser n mayor o igual a m, podemos conjeturar que la longitud de los datos de entrada va a estar dada por la cantidad de dígitos que tiene n. Luego, al medir el tamaño de entrada en términos de d, siendo ésta la cantidad de dígitos de n y, teniendo en cuenta que la entrada se asume en sistema decimal n puede ser representado como $10^d$.

Como habíamos concluido anteriormente que el algoritmo es del orden $\Theta(lg(k))$, luego podemos reemplazar k por $10^d$ y obtenemos $lg(10^d) = d \cdot lg(10)$. Con lo cual, como conclusión podemos determinar que el algoritmo es de orden lineal $\Theta(d)$.