

# Laboratorio Nro. 1

## Recursión

**Katherine Benjumea Ortiz**

Universidad Eafit

Medellín, Colombia

kbenjumeao@eafit.edu.co

**Valentina Ochoa Arboleda**

Universidad Eafit

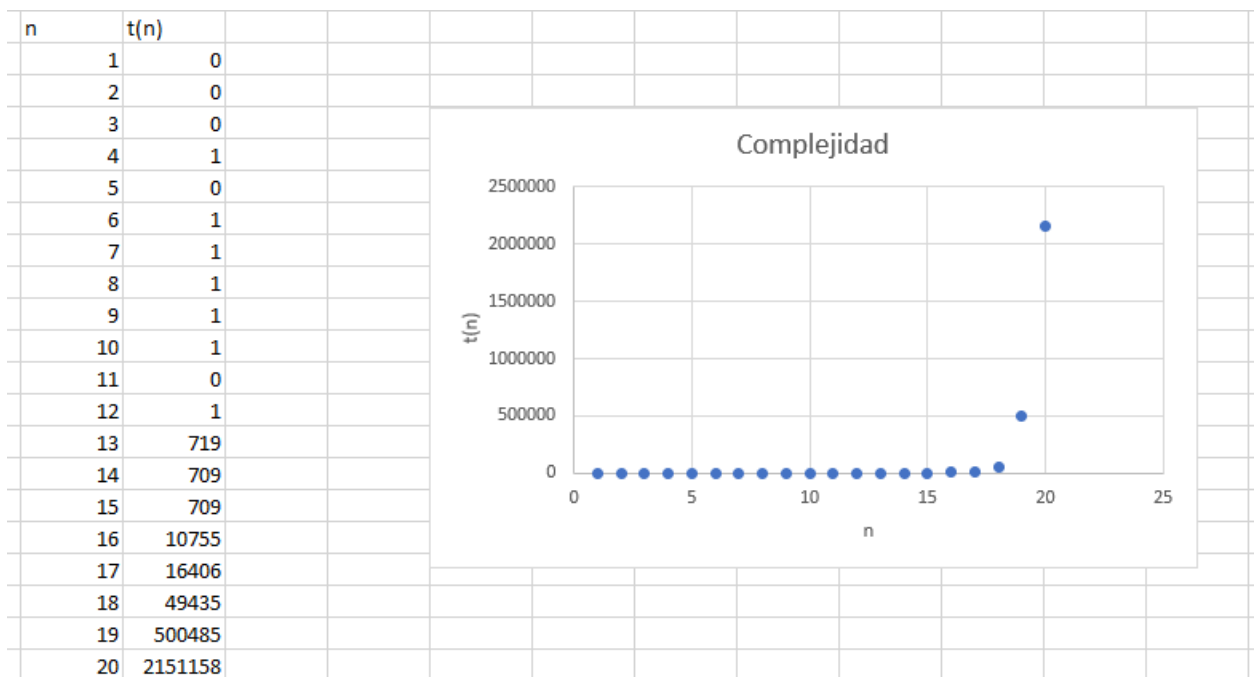
Medellín, Colombia

vochoaa@eafit.edu.co

### 3) Simulacro de preguntas de sustentación de Proyectos

#### 3.1 La complejidad para el algoritmo es $O(4^n)$

#### 3.2



**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245

Los resultados obtenidos en la toma de tiempos nos dejan en evidencia que, aunque el algoritmo funciona adecuadamente, tarda significativamente cuando se le ingresa una longitud de cadenas mitocondriales más altas. Por lo tanto, entre más grande sea la longitud de la cadena más tiempo va a tardar en ejecutarse esto haciendo que su complejidad no sea la apropiada y su eficiencia no sea la esperada.

La estimación en tiempo que se demorará el algoritmo en ejecutarse con dos ADN mitocondriales que tienen un alrededor de 300.000 caracteres es de 27568,565 minutos.

**3.3** La complejidad del algoritmo no es apropiada para los dataset ya que tarda un tiempo significativo para ejecutarse y encontrar la subcadena común entre dos ADN mitocondriales haciendo que el algoritmo no sea eficiente para realizar el proceso esperado.

**3.4** El método de grupSum5 verifica la verdad o falsedad si se le da una matriz de números enteros, puede tomar un grupo de algunos de los enteros para alcanzar un objetivo, el método comienza por el índice o posición de inicio, y también este método tiene un giro que da la instrucción de que todos los múltiplos de cinco deben estar en el grupo y si el valor que sigue inmediatamente a un múltiplo de 5 es 1, este valor no debe ser incluido en el grupo.

### 3.5

#### Recursión 1:

Count7 =  $O(\log(n))$

Count11 =  $O(n)$

Array11 =  $O(n)$

Array6 =  $O(n)$

PowerN =  $O(n)$

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473



## Recursión 2:

GroupNoAdj =  $O(2^n)$

GroupSum6 =  $O(2^n)$

Split53 =  $O(2^n)$

SplitOdd10 =  $O(2^n)$

SplitpArray =  $O(2^n)$

## 3.6

Count7 = n representa el número de entrada

Count11 = n representa el número de entrada

Array6 = n representa el número del tamaño del arreglo

Array 11 = n representa el número de elementos en el array

PowerN = n representa el poder al que se eleva una base

GroupSum6 = n representa el tamaño del arreglo de enteros

GroupNoAdj= n representa el tamaño del arreglo de enteros

Split53 = n representa el tamaño del arreglo de enteros

SplitOdd10 = n representa el tamaño de la matriz de enteros

SplitArray = n representa el número de elementos en el arreglo

## 4) Simulacro de Parcial

### 4.1 A,

#### 4.1.1 A

#### 4.1.2 C

#### 4.1.3 A

### 4.2

4.2.1 *floodFillUtil(screen, x+1, y+1, prevC, newC, N, M);*

*floodFillUtil(screen, x+1, y-1, prevC, newC, N, M);*

4.2.2 *floodFillUtil(screen, x-1, y+1, prevC, newC, N, M);*

*floodFillUtil(screen, x-1, y-1, prevC, newC, N, M);*

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627

Tel: (+57) (4) 261 95 00 Ext. 9473

**ESTRUCTURA DE DATOS 1**  
**Código ST0245**

**4.2.3  $T(p) = p$**

**4.3 B**

**4.4  $lucas(n-1) + lucas(n-2)$**

**4.4.1 C**

**4.5**

**4.5.1 A**

**4.5.2 B**

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

