

## Laboratorio Nro. 1

### Escribir el tema del laboratorio

**Isabella Echeverri Villa**  
Universidad Eafit  
Medellín, Colombia  
iecheverri@eafit.edu.co

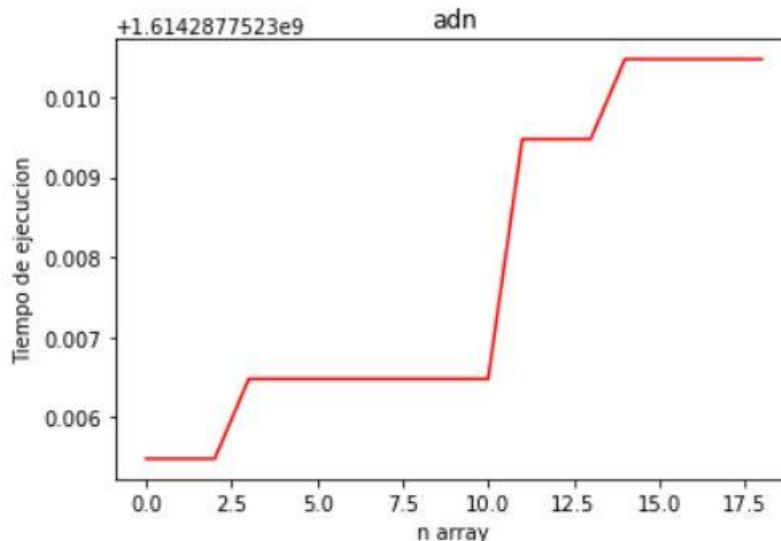
**Samuel Arturo Flórez Rincón**  
Universidad Eafit  
Medellín, Colombia  
saflorezr@eafit.edu.co

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

3.1 La complejidad del ejercicio 1.1 seria:

$$T(n,m)=c1+T(n-1,m-1)+c2+c3+T(n-1,m)+T(n,m-1)$$

3.2



3.3 La complejidad del algoritmo 1.1 es buena para encontrar la subsecuencia común entre dos cadenas de adn pues, al ser cadenas muy largas las cadenas de adn, se necesita un algoritmo eficiente en términos de complejidad y, aunque este algoritmo tiene una complejidad lineal (que no es la mejor de todas) es una complejidad bastante eficiente que sin duda funciona bien para los datasets de adn.

3.4 El ejercicio GroupSum5 funciona de esta manera:

Recibe tres parámetros: una variable start (que es como un contador para irse desplazando en el arreglo), un arreglo y un objetivo o target. Su condición de parada es que la variable

## ESTRUCTURA DE DATOS 1

### Código ST0245

start sea mayor o igual a 5, es decir cuando start sea mayor o igual a 5, el método GroupSum5 va a comenzar a realizar el retroceso en la pila de llamados que se realizaron anteriormente, esta condición de parada debe ser así porque, en caso de que pongamos otra, tendríamos, o el error de ArrayOutOfIndex, porque start seguiría recorriendo el arreglo con un índice mayor a la longitud del arreglo, o el error de no contar algunos números del arreglo.

Posteriormente tenemos el caso recursivo, o la condición recursiva. Como en el enunciado del ejercicio tenemos una especificación sobre los 5 en el arreglo, pusimos un if para asegurarnos de que se cumpliera eso en la ejecución del método. Después de ese if encontramos el caso recursivo que consiste en llamar al mismo método pero con los parámetros start+1 (con el fin de recorrer la siguiente posición del arreglo), el mismo arreglo y el target menos el número que se encuentra en la posición start del arreglo (esto con el fin de saber si hay un conjunto de números (dentro del arreglo) que sumados, con las condiciones específicas, den el target. Luego se procede a verificar si el número siguiente es 1 (ya que hace parte de las especificaciones del problema) y si es el caso, se ajustan los parámetros de manera que ese 1 no cuente dentro del conjunto de números que sumados podrían dar el target. En la última parte, hicimos el caso recursivo basados en el problema original sin las indicaciones específicas de este ejercicio.

### 3.5 Complejidad ejercicios CodingBat Recursion 1: Y 3.6

#### SumDigits

$$T(n) = c3\log_{10}n + c1$$

$$O(c3\log_{10}n + c1)$$

$$O(c3\log_{10}n)$$

$O(\log_{10}n)$  n es el número al que se le suman los dígitos en el método.

#### Triangle

$$T(n) = c3n + c1$$

$$O(c3n + 1)$$

$$O(c3n)$$

$O(n)$  n es el número de filas del triángulo

#### BunnyEars2

$$T(n) = c5n + c1$$

$$O(c5n + c1)$$

$$O(c5n)$$

$O(n)$  n es el número de conejos a los que se le cuentan las orejas.

#### Fibonacci

$$T(n) = c4 + T(n-1) + T(n-2)$$

$$T(n) = c2^n + c$$

$$O(c2^n + c)$$

$$O(c2^n)$$

$O(2^n)$  n es la posición del término que se quiere hallar en la serie de Fibonacci

**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

#### Count7

$$T(n) = c_4 \log_{10} n + c_1$$

$$O(c_4 \log_{10} n + c_1)$$

$$O(c_4 \log_{10} n)$$

$O(\log_{10} n)$  n es el numero al que se le cuentan los 7

#### Ejercicios recursión 2

##### GroupSum6

$$T(n) = t(n+1) + c_5$$

En wólfam:  $T(n) = c_1 - c_5 n$

$$O(c_1 - c_5 n)$$

$$O(c_5 n)$$

$O(n)$  donde n seria la variable "start", start aumenta en 1 cada vez que se hace llamado a la función recursiva, es la variable que siempre está cambiando, a diferencia de la longitud del arreglo y de target (que solo cambia en ciertos casos, no siempre)

##### GroupSum5

$$T(n) = t(n+1) + c_5$$

EN WOLFRAM:  $T(n) = c_1 - c_7 n$

$$O(c_1 - c_7 n)$$

$$O(c_7 n)$$

$O(n)$  donde n seria la variable "start", start aumenta en 1 cada vez que se hace llamado a la función recursiva, es la variable que siempre está cambiando, a diferencia de la longitud del arreglo y de target (que solo cambia en ciertos casos, no siempre)

##### GroupNoAdj

$$\text{Peor caso: } T(n) = c_4 + t(n+2)$$

$$\text{En wólfam: } T(n) = 1/4(c_4((-1)^{2n} - 2n) + c_2(-1)^n + c_1)$$

$$O(1/4(c_4((-1)^{2n} - 2n) + c_2(-1)^n + c_1))$$

$$O(1/4(c_4((-1)^{2n} - 2n) + c_2(-1)^n))$$

$$O(1/4(c_4((-1)^{2n} - 2n)))$$

$$O(1/4(c_4((-1)^{2n}))$$

$$O(1/4((-1)^{2n}))$$

$O((-1)^{2n})$  donde n es la variable que aumenta cada vez que se llama a la función recursivaD

##### Split53

$$\text{Peor caso: } T(n) = t(n+1) + c_8$$

$$\text{WOLFRAM: } T(n) = c_1 - c_8 n$$

$$O(c_1 - c_8 n)$$

$$O(c_8 n)$$

$O(n)$  donde n es "index", que cambia cada vez que se llama al método recursivo

##### SplitArray

$$\text{Peor caso: } T(n) = t(n+1) + c_3$$

$$\text{WOLFRAM: } T(n) = c_1 - c_3 n$$

$$O(c_1 - c_3 n)$$

**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

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

$O(c3n)$

$O(n)$  donde  $n$  es la variable que aumenta en 1 cada que se llama al método recursivo

**4) Simulacro de Parcial**

- 4.1** 1. A.  $s.substring(0,i)$   
2. C.  $true$   
3. A.  $solve(t,s.substring(i),n-i)$

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

**4.3.b.**  $T(n,m)=c*n*m^2$

**4.4.**  $lucas(n-1)+lucas(n-2)$   
**4.4.1.c.**  $T(n)=T(n-1)+T(n-2)+c$ , que es  $O(2^n)$

**4.5.1** A.  $true$   
2.  $b.s.charAt(0)=(s.charAt(s.length()-1))$

**4.6 b.**  $T(n)=2*T(n/2)+C$

**5) Lectura recomendada (opcional)**

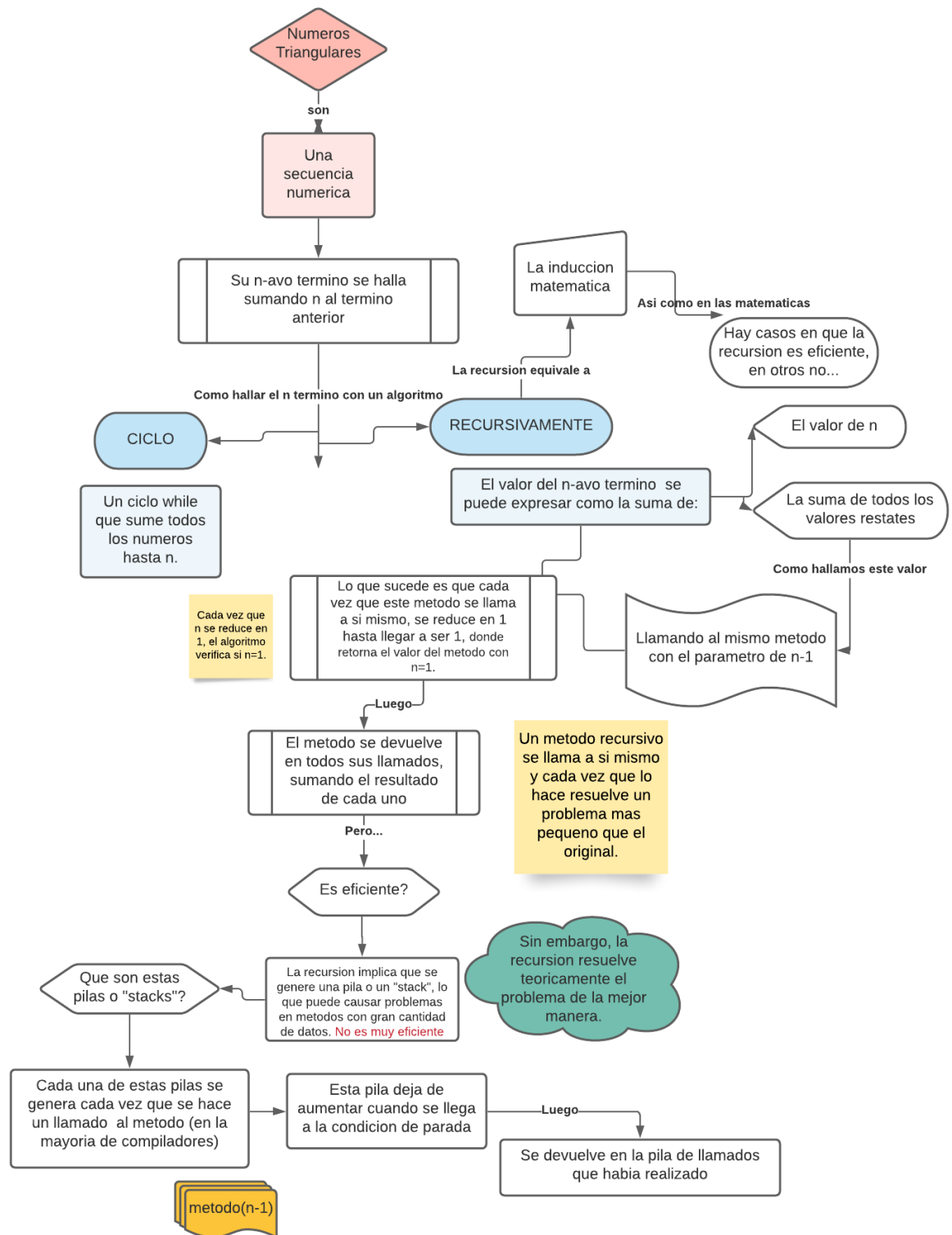
Mapa conceptual

**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



**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