

## Laboratorio Nro. 1

### Escribir el tema del laboratorio

**Jairo Alonso Carvajal**  
Universidad Eafit  
Medellín, Colombia  
correoinegrante1@eafit.edu.co

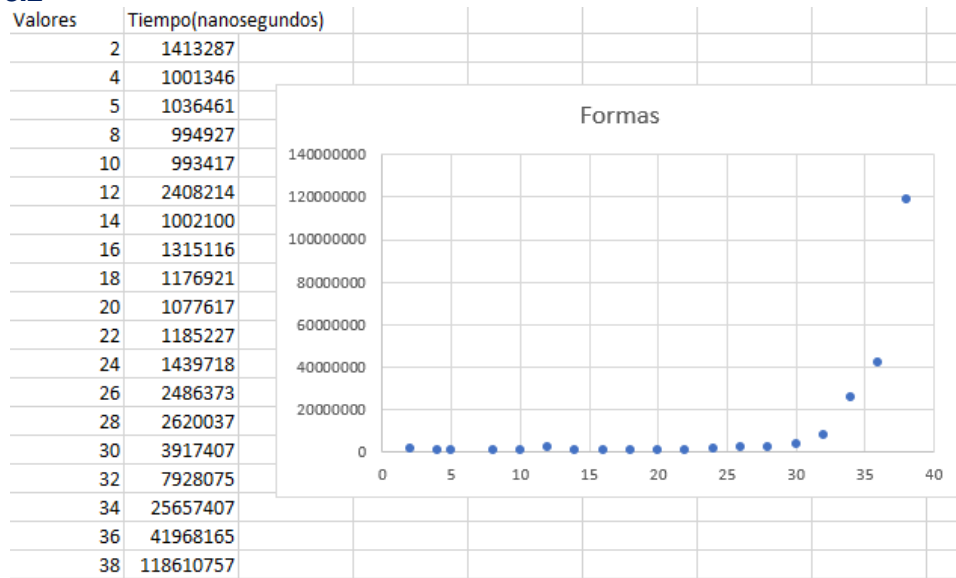
**Alejandra Toro Grisales**  
Universidad Eafit  
Medellín, Colombia  
atorog@eafit.edu.co

```
1) public static int recursion(int n){
    if(n<=2)
        return n;
    return recursion(n-1)+recursion(n-2)/2;
}
```

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

**3.1**  $T(n) = T(n-1) + T(n-2) + C$

**3.2**



**3.3** La complejidad que presenta este algoritmo, la cual se definiría por una ecuación exponencial, hace que no sea viable utilizarlo en Puerto Antioquia, en 2020, con contenedores que miden miles de centímetros.

**3.4** Dado un arreglo, una posición inicio y un número  $n$ , el algoritmo compara si la suma de algunos de los elementos del arreglo es igual a  $n$ , lo que hace primeramente el algoritmo es verificar si hay algún elemento en el arreglo, luego comprueba si en determinada posición

**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

hay un 5 y si el número que sigue de este es un 1 el algoritmo no lo tiene en cuenta, de ser así pasara a la siguiente posición después del 1 y, si la condición no fuese satisfecha se pasara a la posición inicio+1, por último el algoritmo repite este ciclo sumando 1 a la posición inicio hasta que esta sea mayor a la longitud del arreglo. Ya tendremos un valor boolean guardado, el algoritmo repetirá este ciclo y nos dará otro valor boolean si ambos son cierto retornará true, de no ser así será false.

### 3.5

#### Recursión 1:

<b>Ejercicio</b>	<b>Complejidad</b>
<a href="#"><u>countX</u></a>	$T(n)=c+T(n+1)$ n: en el algoritmo es i y lo que haces es devolver las posiciones de donde se encuentra 'x'
<a href="#"><u>strCount</u></a>	$T(n)=c+T(n+1)$ n: en este caso 1 va sumando la letra que se parezca a sub, y agregandole 1 al contador a ver si cumple con el tamaño de sub.
<a href="#"><u>factorial</u></a>	$T(n)=T(n-1)+c$ n: es el número que se irá multiplicando por su antecedente.
<a href="#"><u>bunnyEars2</u></a>	$T(n)=(2+T(n-1)+(3+T(n-1)))+C$ n: es el número que dependiendo de cierta condición actuará sumando 2 o 3.
<a href="#"><u>count7</u></a>	$T(n)=1+T(n/10)+T(n/10)+C$ n: es el número que dependiendo de cierta condición sumará 1 o no.

#### Recursión 2:

<b>Ejercicio</b>	<b>Complejidad</b>
<a href="#"><u>groupSum</u></a> H	$T(n) = T(n-1)+(n+1)+c$ n: n en el primer llamado recursivo va sumando q paa poder i recorriendo el arreglo. n: en el segundo llamado recursivo lo que hace es pasar a la siguiente posición ignorando la posición que esta ya que no encontró un valor que corresponda.
<a href="#"><u>groupSum6</u></a>	$T(n) = c+2(T(n+1))$ n: equivale al aumento de start para recorrer el arreglo.
<a href="#"><u>groupNoAdj</u></a>	$T(N) = c+T(n+1)+T(n+2)$ N: en el primer llamado recursivo lo que haces pasar posiciones en el arreglo.

**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**

	N: Segundo llamado recursivo lo que hace es pasar de posición y restar lo que hay ahí.
<u>GroupSumClump</u>	$T(n) = T(n+1) + T(n+1) + C$ n: dará la posición que irá recorriendo el arreglo.
<u>GroupSum5</u>	$T(n) = T(n+1) + T(n+1) + C$ n: es el número que dará la posición del arreglo hasta recorrerlo por completo.

#### 4) Simulacro de Parcial

**4.2** La ecuación  $T(n) = T(n-1) + C$  es la que más se adecua al comportamiento del algoritmo 4.2.

**4.3 4.3.1** Línea 4: `int res = solucionar(n-1,a,b,c) + 1;`

**4.3.2** Línea 5: `res = Math.max(a,b);`

**4.3.3** Línea 6: `res = Math.max(res,c);`

**4.5 4.5.1** Línea 2: `return n;`

**4.5.2** Línea 3: `formas(n-1);`

**4.5.3** Línea 4: `formas(n-2)/2;`

**4.5.2**  $T(n) = T(n-1) + T(n-2) + C$

**4.6.1** Línea 10: `sumaAux(n,i+1);`

**4.6.2** Línea 12: `sumaAux(n,i+1);`

**4.8.1** Línea 9: `return 0;`

**4.8.2** Línea 13: `int suma=nj+ni ;`

**4.10** La salida que se esperaría cuando fun es igual a fun(1,4) sería 6.

**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