

## Laboratorio Nro. X

### Escribir el tema del laboratorio

**Andrés Ospina Patiño**  
Universidad Eafit  
Medellín, Colombia  
aospinap1@eafit.edu.co

**Felipe Álvarez Benítez**  
Universidad Eafit  
Medellín, Colombia  
falvarezb@eafit.edu.co

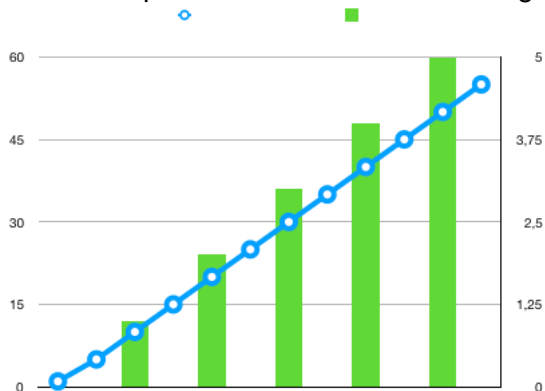
### 3) Simulacro de preguntas de sustentación de Proyectos

**3.1** La complejidad asintótica del problema de como acomodar los rectángulos es de  $O(2n)$  su ecuación de recurrencia es  $T(n)=T(n-1)+T(n-2)$ .

**3.2** Ejecutando el código:

```
long start = System.currentTimeMillis ();
nRectangulos (50);
long end = System.currentTimeMillis ();
System.out.println (end -start);
```

Tenemos que se demora 126265 milisegundos, lo que equivale a 126.265 segundos.



**3.3** Podría ser implementado en puerto Antioquia en 2020 debido a que es un algoritmo, pero hay que tener en cuenta que acá se está trabajando con dos dimensiones, por lo que este tendría que sufrir algunas modificaciones para poder utilizarlo con volúmenes.

**3.4** GroupSum5 tiene el mismo principio de un GroupSum normal, sin embargo en este hay que considerar varios eventos, siempre que sea múltiplo de 5 se debe sumar y si hay un 1 inmediatamente después de este no puede ser sumado.

**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

Como condición de parada se tiene que si “start” es mayor o igual al tamaño del arreglo retorne verdadero solo si target es igual a 0, sino el programa deberá ir sumando al “start” para que este sea igual al tamaño del arreglo; si el número es múltiplo de 5 se le resta este al target y se le suma uno al “start”, a menos de que el número inmediatamente siguiente sea 1, en este caso se le debrá sumar 2 al “start” para que este no sea considerado.

### 3.5

#### Recursión 1:

- Factorial:  $T(n) = n * T(n-1)$
- bunnyEars:  $T(n) = 2 + T(n-1) + C$
- Fibonacci:  $T(n) = T(n-1) + T(n-2) + C$
- bunnyEars2:  $T(n) = c + T(n-1) + C$
- triangle:  $T(n) = c + T(n-1) + C$

#### Recursión 2:

- groupSum:  $T(n) = 2T(n-1) + C$
- groupSum6:  $T(n) = 2T(n-1) + C$
- groupSumNoAdj:  $T(n) = T(n-1) + T(n-2) + C$
- groupSum5:  $T(n) = 2T(n-1) + C$
- splitArray:  $T(n) = 2T(n-1) + C$

**3.6** Las variables n, c presentes en los diferentes ejercicios representan el número de elementos en el arreglo y una constante respectivamente.

## 4) Simulacro de Parcial

**4.1** start+1, nums, target

**4.2** a.  $T(n/2) + c$ .

**4.3**

- n-a, a, b, c
- res, solucionar (n-b, a, b, c)+1
- res, solucionar (n-c, a, b, c)+1

**4.4** e.

**4.5**

- return n
- n-1
- n-2

En el peor de los casos el algoritmo ejecuta  $T(n)=T(n-1)+T(n-2)+c$

**4.6**

- sumaAux(n, i+2)
- sumaAux(n.substring(i+1), i)

**4.7**

- S, i+1, t
- S, i+1, t-S[i]

**4.8**

**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

- *return 0*
- *return ni+nj*

**4.9** *c*

**4.10** *b*

**4.11**

- *n-1*
- *lucas(n-2)*

**4.12**

- *return sat*
- *sat += Math.max(fi, fj)*
- *return sat*

### 5) Lectura recomendada (opcional)

Mapa conceptual

### 6) Trabajo en Equipo y Progreso Gradual (Opcional)

**6.1** *Actas de reunión*

**6.2** *El reporte de cambios en el código*

**6.3** *El reporte de cambios del informe de laboratorio*

**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

