

## Laboratorio Nro. 1

**Santiago Vanegas Higueta**  
Universidad Eafit  
Medellín, Colombia  
svanegash@eafit.edu.co

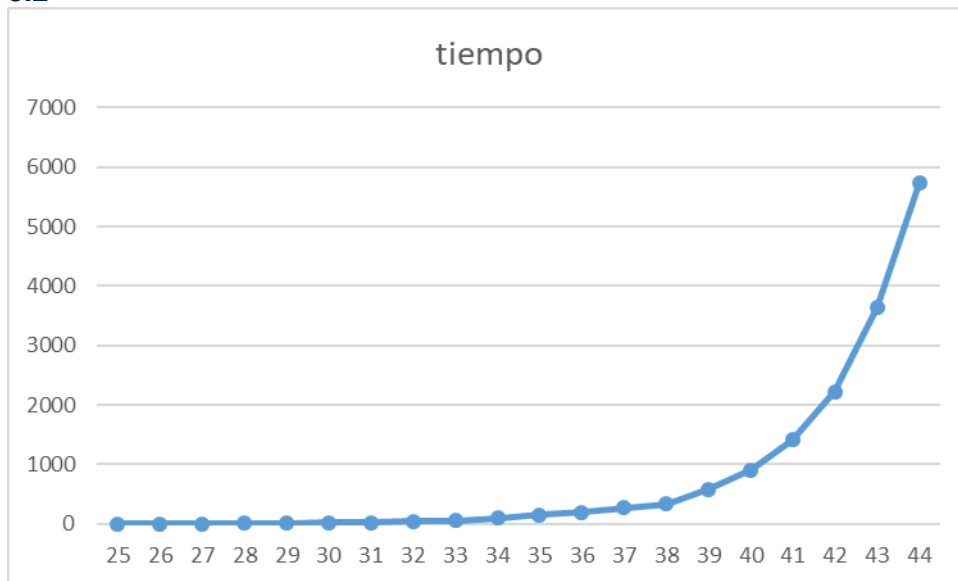
**Juan Camilo Iguaran Fernandez**  
Universidad Eafit  
Medellín, Colombia  
Jciguaranf@eafit.edu.co

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

#### 3.1

$$t(n) = C1 + C2 + C3 + C4 + C5(t(n-2) + t(n-3))$$

#### 3.2



El tiempo que demoraría cuando el contenedor midiera 50x2cm<sup>2</sup> sería de 63857.

#### 3.3

## ESTRUCTURA DE DATOS 1

### Código ST0245

No me parece un buen algoritmo puesto que si eso es lo que demora con un contenedor tan pequeño, debe demorar demasiado con un contenedor de miles de centímetros cúbicos.

#### 3.4

El código groupsum5 será el encargado de ver si existiera alguna forma de sumar unos números, y esa suma debe ser igual  $x$  número, es decir, al target.

Se tienen dos restricciones: todos los múltiplos de 5 en el arreglo deben ser escogidos y si después de ese múltiplo de 5 le sigue un 1, este 1 se omite.

La primera parte del código emplea una condición: que si start es mayor a la longitud del arreglo se vaya a otra condicional: que si target es igual a 0 retorne verdadero, si no retorne verdadero.

Al salirse de ese condicional pregunta que si nums en la posición start es múltiplo de 5 se vaya a otra condición: si start es menor a la longitud y nums en la posición start más uno es igual a 1 entonces se le sumará 2 a start y se restará target - nums en la posición start; todo de manera recursiva.

Luego retornará la suma de 1 a start y la resta de target - nums en la posición start, de manera recursiva también.

Luego se irá a otra condición en la cual se le sumará 1 a start y se restará target - nums en la posición start, de manera recursiva. Si esto es verdad se retorna verdadero.

Y en la última línea se le sumará 1 a start de manera recursiva.

#### 3.5

```
public int factorial(int n) {
    if(n<=1)//C1
        return 1;//C2
    else
        return n * factorial(n-1);//C3 * T(n-1)
    //T(n)= C1+C2+C3 * (n-1)
}

public int bunnyEars(int bunnies) {
    if(bunnies == 0)//C1
        return 0;//C2
    else
        return 2 + bunnyEars(bunnies-1);//C3+T(n-1)
    //T(n)= C1+C2+C3
}

public int fibonacci(int n) {
    if(n==0)//C1
        return 0;//C2
```

## ESTRUCTURA DE DATOS 1

### Código ST0245

```

else if(n==1)//C3
return 1;//C4
else
return fibonacci(n-2) + fibonacci(n-1);//T(n-2)+T(n-1)
//T(n)=C1+C2+C3+C4+C5(t(n-2) + t(n-1))
}

public int bunnyEars2(int bunnies) {
    if (bunnies == 0)//C1
        return 0;//C2
    else if(bunnies == 1)//C3
        return 2;//C4
    else if(bunnies%2==1)//C5
        return 2 + bunnyEars2(bunnies-1);//C6 + T(n-1)
    else
        return (3 + bunnyEars2(bunnies-1));//C7+T(n-1)
    //T(n)= C1+C2+C3+C4+C5+C6+C7
}

public int triangle(int rows) {
    if(rows==0)//C1
        return 0;//C2
    else
        return rows + triangle(rows-1);//C3 + T(n-1)
    //T(n)= C1+C2+C3
}

```

### 3.6

n es el número con el que se quiere trabajar  
 bunnies es el número de bunnies  
 rows es el numero de rows  
 start es el comienzo de una cadena que siempre es cero  
 nums es el numero de caracteres  
 target es el numero al que quiero llegar  
 La i es el número de iteraciones  
 Y el balance es para dividir pues las parejas de números

## 4) Simulacro de Parcial

### 4.1 a

### 4.2 A

4.3.1) (n-1, a ,b, c)

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

### 4.3

4.3.1) (n-1, a ,b, c)

## ESTRUCTURA DE DATOS 1

### Código ST0245

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

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

**4.4**

**4.5**

4.5.1) Línea 2: return n

Línea 3: n-1

Línea 4: n-2

4.5.2) b

4.6.1)  $i=i+2$

4.6.2) sumaAux(n,i+1)

4.8.1) return 0;

4.8.2)  $n_i + n_j$

4.10) b

### 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