

Laboratorio Nro. 1 Recursión

Camilo Villa Tamayo
Universidad Eafit
Medellín, Colombia
cvillat@eafit.edu.co

Miguel Ángel Sarmiento Aguiar
Universidad Eafit
Medellín, Colombia
msarmie4@eafit.edu.co

Marlon Pérez Ríos
Universidad Eafit
Medellín, Colombia
mperezr@eafit.edu.co

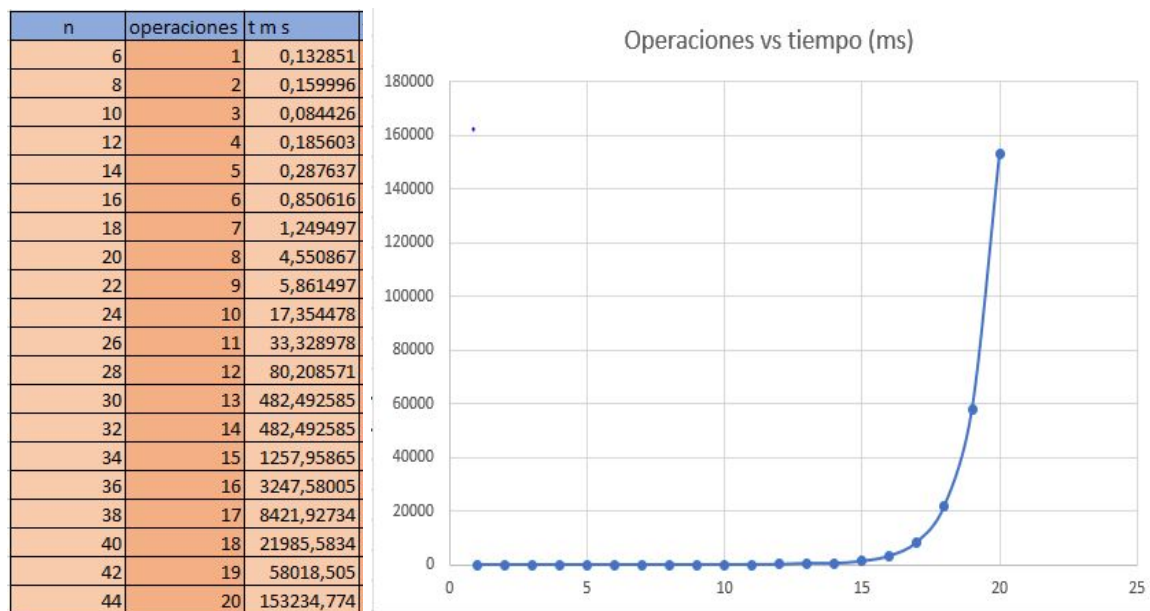
3) Simulacro de preguntas de sustentación de Proyectos

3.1 La complejidad asintótica de este ejercicio es:

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

$$T(n) = O(2^n)$$

3.2



3.3 Al ser contenedores de miles de centímetros aumenta la n y así mismo la complejidad del algoritmo, siendo esto un aumento de tiempo en la ejecución sin llegar a una solución óptima en tiempo.

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

3.5 3.5.1

- **Factorial:** $T(n) = T(n-1) + C$
- **Fibonacci:** $T(n) = T(n-1) + T(n-2) + C$
- **BunnyEars:** $T(n) = T(n-1) + C$
- **Triangle:** $T(n) = T(n-1) + C$
- **SumDigits:** $T(n) = T(n-1) + C$

3.5.2

- **SplitArray:** $T(n) = T(n-1) + T(n-1) + C$
- **SplitOdd10:** $T(n) = T(n-1) + T(n-1) + C$
- **Split53:** $T(n) = T(n-1) + T(n-1) + T(n-1) + T(n-1) + C$
- **GroupSumClump:** $T(n) = T(n-1) + T(n-1) + C$
- **GroupSum6:** $T(n) = T(n-1) + T(n-1) + T(n-1) + C$

3.6 3.6.1

- **Factorial:** n es el número al que se le aplica el factorial.
- **Fibonacci:** n es el n-ésimo número de la serie.
- **BunnyEars:** n es el número de conejos.
- **Triangle:** n es el número de filas del triángulo.
- **SumDigits:** n es el número al que se le quieren sumar sus dígitos.

3.6.2

- **SplitArray:** n es el arreglo de números.
- **SplitOdd10:** n es el arreglo de números.
- **Split53:** n es el arreglo de números.
- **GroupSumClump:** n es el arreglo de números.
- **GroupSum6:** n es el arreglo de números.

4) Simulacro de Parcial

4.1 LÍNEA 3: return True

LÍNEA 4: if (s[0] == s[len(s)-1]):

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

4.3 4.3.1 LÍNEA 4: res = solucionar(n-1, a, b, c) + 1

4.3.2 LÍNEA 5: res = max(solucionar(n-a, a, b, c), solucionar(n-b, a, b, c))

4.3.3 LÍNEA 6: res = max((n-res), solucionar(n-c, a, b, c))

4.4 e. La suma de los elementos del arreglo a y es $O(n)$

4.5 4.5.1 LÍNEA 3: if (T == 0) return 1

LÍNEA 4: elif (T < 0) return 0

LÍNEA 8: return f1+f2+f3

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

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

- 4.6** **4.6.1** LÍNEA 10: return sumAux(n[i+1:], i+1)
 4.6.2 LÍNEA 12: return int(n[i]) + sumAux(n[i+1:], i+1)
- 4.7** **4.7.1** LÍNEA 9: return comb(S, i+1, t-S[i]) or
 4.7.2 LÍNEA 10: comb(S, i+1, t)
- 4.8** **4.8.1** LÍNEA 9: return 0
 4.8.2 LÍNEA 13: suma = ni+nj
- 4.9** c. 22
- 4.10** b. 6
- 4.11** **4.11.1** LÍNEA 4: return lucas(n-1) + lucas(n-2)
 4.11.2 c. $T(n) = T(n-1) + T(n-2) + C$, que es $O(2^n)$
- 4.12** **4.12.1** LÍNEA 13: return sat
 4.12.2 LÍNEA 17: sat += max(fi, fj)
 4.12.3 LÍNEA 18: return sat

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

