

Laboratorio Nro. 1

Recursión-Complejidad

Jaime Rodrigo Uribe Mogollon

Universidad Eafit
Medellín, Colombia
jruribem@eafit.edu.co

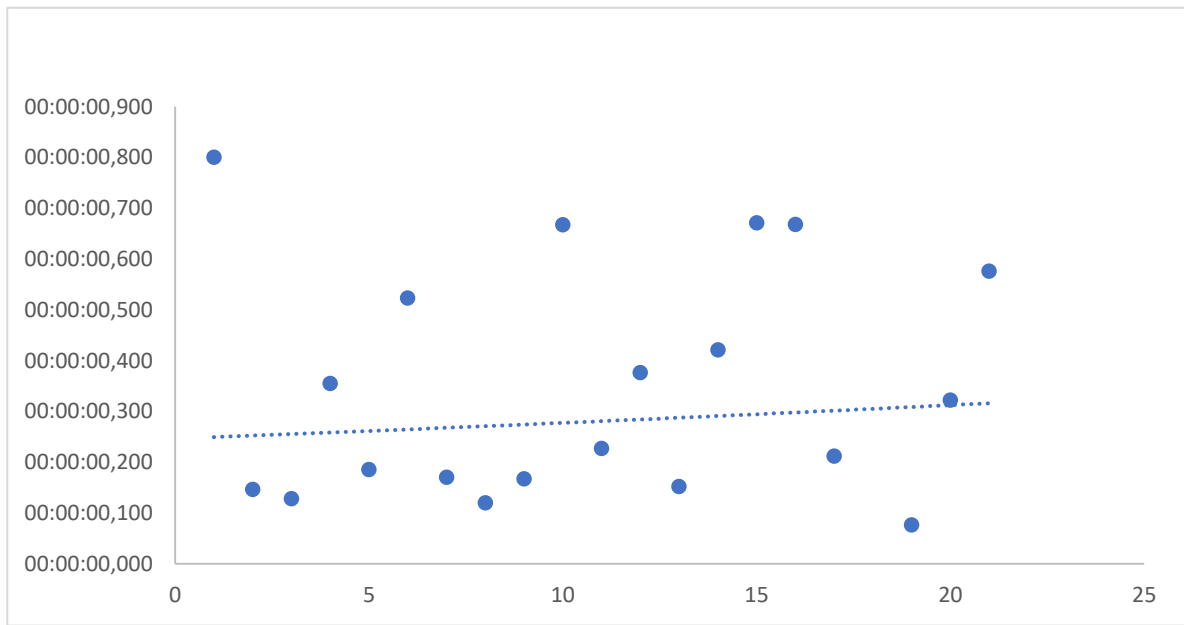
Daniel Alberto Posada Murillo

Universidad Eafit
Medellín, Colombia
Dposad21@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

3.2



El tiempo total fue de 800 ms

El tiempo total fue de 146 ms

El tiempo total fue de 128 ms

El tiempo total fue de 355 ms

El tiempo total fue de 185 ms

El tiempo total fue de 523 ms

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas

ESTRUCTURA DE DATOS 1

Código ST0245

El tiempo total fue de 1705 ms
 El tiempo total fue de 120 ms
 El tiempo total fue de 1676 ms
 El tiempo total fue de 667 ms
 El tiempo total fue de 2275 ms
 El tiempo total fue de 376 ms
 El tiempo total fue de 1525 ms
 El tiempo total fue de 421 ms
 El tiempo total fue de 671 ms
 El tiempo total fue de 668 ms
 El tiempo total fue de 212 ms
 El tiempo total fue de 76 ms
 El tiempo total fue de 322 ms
 El tiempo total fue de 576 ms

3.3 Al observar el comportamiento del algoritmo, se asemeja al de una función exponencial, por lo cual tendríamos problemas al evaluar cadenas de gran tamaño.

3.4 En este algoritmo, cada vez que se hace, cuenta un elemento menos de la matriz y se tiene en cuenta o no un elemento más de la matriz dependiendo del caso y llega hasta que el parámetro de inicio llega al final de la matriz. Si se da el caso, que el parámetro de inicio es múltiplo de 5 y el elemento que sigue en la matriz es 1, se hace un llamado recursivo omitiendo dos elementos debido a que se hace uso del `start+2`. Si este no es el caso, se invoca `star+1`. Este algoritmo tiene como finalidad devolver un booleano verdadero si es posible que se forme con la suma el parámetro de destino y un falso si esto no se cumple.

3.5

Recursión 1

- BunnyEars2 = $T(n-1) + C$
 $O(n)$
- Sumdigits = $T(n/10) + c$
 $O(\log(n))$
- Count7 = $T(n/10) + c$
 $O(\log(n))$
- PowerN = $T(n-1) + C$
 $O(n)$
- Array11 = $T(n-1) + C$

ESTRUCTURA DE DATOS 1

Código ST0245

$O(n)$

Recursión 2

- GroupSum6 = $T(n-1) + T(n-1) + C$ $O(2^n)$
- GroupNoAdj = $T(n-1) + C$
 $O(n)$
- Split53 = $T(n-1) + T(n-1) + C$ $O(2^n)$
- SplitOdd10 = $2 * T(n-1) + C$
 $O(2^n)$

3.6 BunnyEars2: n es el número de conejos que debemos analizar.

Sumdigits: n representa el dígito a evaluar.

Count7: n representa el dígito a evaluar

PowerN: en la expresión, n representa el exponente.

Array11: n vendría siendo la cantidad de elementos del arreglo

GroupSum6: n representa el número de elementos del arreglo, haciendo un recorrido con Start al cual, en cada llamado, se le va adicionando una unidad.

GroupNoAdj: n representa el número de elementos del arreglo, haciendo un recorrido con Start al cual se le adiciona 2 si se le resta el elemento de la posición en la que se encuentra al número el cual se desea llegar.

Split53: n representa el número de elementos del arreglo, donde en cada llamado, se suma una unidad a index y es usada para crear una nueva variable

SplitOdd10: n representa el número de elementos del arreglo, haciendo un recorrido con i a la cual se le va sumando 1.

ESTRUCTURA DE DATOS 1

Código ST0245

4) Simulacro de Parcial

4.1

Línea 6: C línea

9: C línea 11: A

4.3

B

4.4

C

4.5 línea

3: A línea

4: 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