

Laboratorio Nro. 1 Recursión

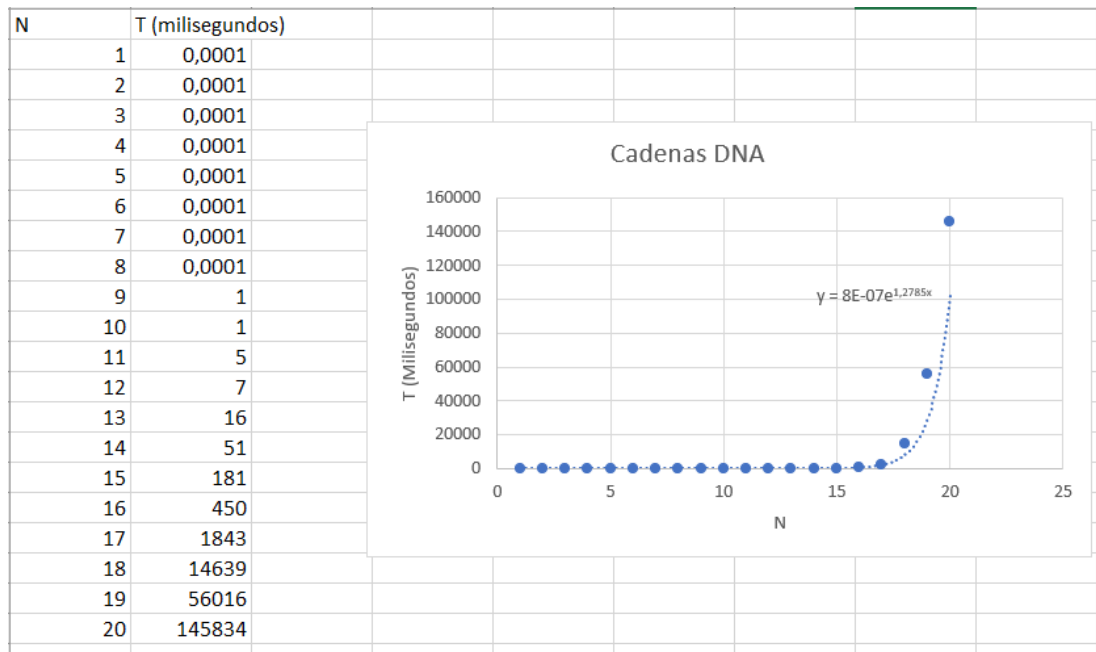
Juan Pablo Yepes
Universidad Eafit
Medellín, Colombia
jpyepesg@eafit.edu.co

Simón Cárdenas Villada
Universidad Eafit
Medellín, Colombia
scardenasv@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1
$$t(n) = \frac{1}{2} (c_1 + c_2 + c_3) (3^n - 1) + c_1 3^{n-1}$$
 (where c_1 is an arbitrary parameter)

3.2



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

Para el caso de los 300000 caracteres se reemplazaría en la función:

$$8E - 07e^{1.2785 \cdot (300000)}$$

Esto da como resultado un número tan grande que no puede ser calculado por una calculadora moderna.

3.3

Este algoritmo no es apropiado para implementarlo en el ejercicio de ADN ya que dicho algoritmo tiende a ser exponencial (en cuestiones de tiempo de ejecución) por lo tanto, mientras más grande sea la cadena a comparar más se demorará en procesar una solución.

3.4 Explicación de groupSum5

Casos base: Primero hay que iterar por todo el arreglo para determinar que, si se hayan analizado todos sus elementos, luego se analiza si el target al que se quiere llegar es igual a cero para retornar que si es posible hallar la suma.

Caso recursivo: Primero se revisa si el valor actual es múltiplo de 5, luego se revisa si el valor que sigue es igual a 1. Si estas dos condiciones son verdaderas, se llama la función con el valor actual y se salta el valor siguiente. Si el valor que sigue no es 1, se llama la función usando el múltiplo de 5 sin saltarse ningún valor. Por último, está el caso si el valor actual no es un múltiplo de 5, donde se decide si se usa el valor actual o no.

3.5

Recursión 1:

Factorial:

$$t(n) = c_1 + t(n - 1)$$

BunnyEars:

$$t(n) = (c_1 + c_2)n + c_1 \text{ (where } c_1 \text{ is an arbitrary parameter)}$$

Fibonacci:

$$t(n) = -c_1 - c_2 - c_3 + c_1 F_n + c_2 L_n \text{ (where } c_1 \text{ and } c_2 \text{ are arbitrary parameters)}$$

F_n is the n^{th} Fibonacci number

BunnyEars2:

$$t(n) = (c_1 + c_2 + c_3)n + c_1 \text{ (where } c_1 \text{ is an arbitrary parameter)}$$

Triangle:

$$t(n) = (c_1 + c_2 + c_3)n + c_1 + \frac{1}{2}n(n + 1) \text{ (where } c_1 \text{ is an arbitrary parameter)}$$

Recursion 2:

groupSum6:

$$t(n) = \frac{1}{2}(-c_1 - c_2 - c_3) + c_1(-1 - \sqrt{2})^n + c_2(\sqrt{2} - 1)^n$$

(where c_1 and c_2 are arbitrary parameters)

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

groupNoAdj:

$$t(n) = c_1 - (c_1 + c_2 + c_3) n \quad (\text{where } c_1 \text{ is an arbitrary parameter})$$

groupSum5:

$$t(n) = \frac{1}{3} (-c_1 - c_2 - c_3 - c_4 - c_5 - c_6) + c_1 \left(\frac{1}{2} (-3 - \sqrt{13}) \right)^n + c_2 \left(\frac{1}{2} (\sqrt{13} - 3) \right)^n$$

(where c_1 and c_2 are arbitrary parameters)

groupSumClump:

$$t(n) = c_1 - (c_1 + c_2) n \quad (\text{where } c_1 \text{ is an arbitrary parameter})$$

splitArray:

$$t(n) = 2^{-n} (2c_1 - (c_1 + c_2 + c_3) (2^n - 1)) \quad (\text{where } c_1 \text{ is an arbitrary parameter})$$

String 1 (cadena) es la variable de longitud m, String 2 (cadena) es la variable de longitud n.

4) Simulacro de Parcial

4.1

1. a. s.substring(0,i)
2. c. true
3. a. solve(t, s.substring(i), n-i)
- 4.

4.2

1. (a) verdadero.
2. (a),(c)

4.3 b. $T(n,m) = C * n * m^2$

4.4 return lucas (n-1) + lucas (n-2);

4.4.1. $t(n) = t(n-1) + t(n-2) + c$, que es $O(2^n)$

4.5 a. true

b. $s.charAt(0) == (s.charAt(s.length()-1))$

4.5.1. a. $t(n) = t(n-1) + c$

4.6

4.6.1 return sumaAux(n,i+2)

4.6.2 sumasAux(n, i+1)

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