

ESTRUCTURA DE DATOS 1

Código ST0245

Laboratorio Nro. 1

Recursión

Laura Katterine Zapata Rendón

Universidad Eafit

Medellín, Colombia

lkzapatar@eafit.edu.co

Maria Alejandra Vélez Clavijo

Universidad Eafit

Medellín, Colombia

mavelezc1@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Tenemos que **m** es el número de caracteres de la cadena 1 y **n** es el número de caracteres de la cadena 2, luego **p** es la suma de estas dos longitudes, así: $p = m + n$.

La complejidad asintótica para el peor de los casos del ejercicio 1.1 es :

$$T(p) = c_3 + T(p-1) + T(p-1), \text{ donde } c_3 = 4$$

$$T(p) = c_3(2^p - 1) + c_1 * 2^{p-1}$$

$$T(p) = (c_3 + \frac{c_1}{2}) * 2^p - c_3$$

$$T(p) = O((c_3 + \frac{c_1}{2}) * 2^p - c_3) \text{ (Simetría)}$$

$$O((c_3 + \frac{c_1}{2}) * 2^p - c_3) = O((c_3 + \frac{c_1}{2}) * 2^p) \text{ (regla de la suma)}$$

$$O((c_3 + \frac{c_1}{2}) * 2^p) = O(2^p) \text{ (regla del producto)}$$

ESTRUCTURA DE DATOS 1

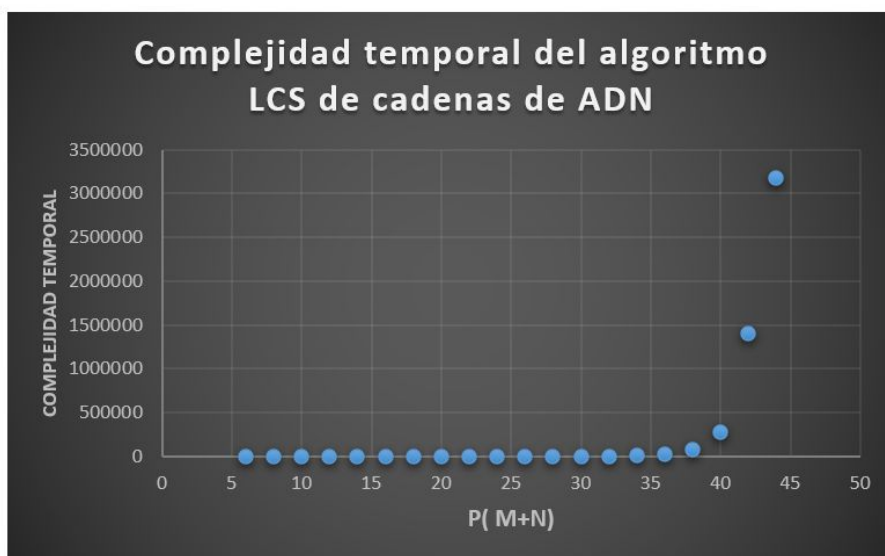
Código ST0245

$T(n)$ es $O(2^p)$ es decir que la complejidad asintótica es exponencial.

3.2 La complejidad del algoritmo para encontrar la subsecuencia en común más larga entre dos cadenas de ADN mitocondrial con 300.000 caracteres cada una, es de 2^p donde p es la suma de los caracteres de ambas cadenas es decir $p = m + n$ donde $m = 300.000$ y $n = 300.000$. Luego $p = 600.000$ y el tiempo que se tarda en encontrar dicha subsecuencia es $2^{600.000}$.

Analizando el tiempo que demora el algoritmo en ejecutarse para cuando cada cadena tiene 20 caracteres se deben hacer 1099511627776 operaciones y esto toma alrededor de 8.7 minutos. Ahora con cadenas de 300.000 caracteres como las de los datasets, sería una cantidad asintótica de operaciones y tomaría aproximadamente 3 meses en ejecutarse dependiendo de la capacidad del pc.

P(m+n)	O(2 ^p)
6	0
8	0
10	0
12	0
14	0
16	1
18	1
20	1
22	4
24	20
26	38
28	139
30	588
32	1796
34	4974
36	18149
38	70040
40	275072
42	1401184
44	3180947



Analizando la gráfica de la complejidad temporal para el peor de los casos del algoritmo LCS de cadenas de ADN, es importante resaltar que se inició con dos cadenas de 3

ESTRUCTURA DE DATOS 1

Código ST0245

caracteres y se finalizó con dos cadenas de 22 caracteres cada una, destacando que esta última se demoró 53 minutos en ejecutarse.

De esto puede concluir que la relación entre el tiempo y la cantidad de operaciones que se deben hacer para ejecutar el algoritmo con dos cadenas de m y n caracteres respectivamente, es exponencial.

Entre mayor sea el número de caracteres que tengan nuestras cadenas a analizar mucho más se va a demorar en ejecutarse, por ende, no es recomendable utilizar este algoritmo para cadenas con gran cantidad de caracteres.

3.3 ¿La complejidad del algoritmo del ejercicio 1.1 es apropiada para encontrar la subsecuencia común más larga entre ADNs mitocondriales como los de los datasets?

Dado que la complejidad del algoritmo del ejercicio 1.1 es exponencial 2^p , y teniendo en cuenta que p es la suma de las longitudes de las cadenas, afirmamos que no es apropiado utilizar este algoritmo con esta complejidad para encontrar la secuencia común más larga entre ADNs ya que teniendo en cuenta el tiempo que se demora en ejecutarse con menos de 20 caracteres cada cadena, podemos concluir que si lo ejecutamos con una cantidad de caracteres como los de los datasets tardaría una cantidad exagerada de tiempo.

3.5 . Calculen la complejidad de los Ejercicios en Línea de los numerales 2.1 y 2.2 **y**

3.6 . Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del ejercicio anterior.

Complejidad de ejercicios recursión 1

- **countHi:**

Complejidad para el peor de los casos:

$$t(n) = c_1 + t(n-1), \text{ donde } c_1 = 5$$

$$t(n) = c_3 \cdot n + c_1$$

$$O(c_3 \cdot n + c_1) \text{ (simetría)}$$

$$O(c_3 \cdot n) \text{ (regla de la suma)}$$

$$O(2) \text{ (regla del producto)}$$

ESTRUCTURA DE DATOS 1

Código ST0245

complejidad: $O(n)$ lineal

Donde n es el número de caracteres que tiene la cadena a analizar

- **bunnyEars2:**

Complejidad para el peor de los casos:

$$t(n) = c_1 + t(n-1), \text{ donde } c_1 = 6$$

$$t(n) = c_3 * n + c_1$$

$$O(c_3 * n + c_1) \text{ (Simetría)}$$

$$O(c_3 * n) \text{ (regla de la suma)}$$

$$O(n) \text{ (regla del producto)}$$

complejidad: $O(n)$ lineal

Donde n es el número de conejos

- **triangle:**

Complejidad para el peor de los casos:

$$t(n) = c_1 + t(n-1), \text{ donde } c_1 = 4$$

$$t(n) = c_3 * n + c_1$$

$$O(c_3 * n + c_1) \text{ (Simetría)}$$

$$O(c_3 * n) \text{ (regla de la suma)}$$

$$O(n) \text{ (regla del producto)}$$

complejidad: $O(n)$ lineal

Donde n es el número de filas del triángulo

- **bunnyEars**

Complejidad para el peor de los casos:

ESTRUCTURA DE DATOS 1

Código ST0245

$$T(n) = c_2 + T(n-1), \text{ donde } c_2=3$$

$$T(n) = c_2 * n + c_1$$

$$T(n) = O(c_2 * n + c_1) \text{ (Simetría)}$$

$$O(c_2 * n + c_1) = O(c_2 * n) \text{ (regla de la suma)}$$

$$O(c_2 * n) = O(n) \text{ (regla del producto)}$$

$T(n)$ es $O(n)$ es decir que la complejidad asintótica es lineal.

Donde n es el número de conejos

- factorial

Complejidad para el peor de los casos:

$$T(n) = c_2 + T(n-1), \text{ donde } c_2=3$$

$$T(n) = c_2 * n + c_1$$

$$T(n) = O(c_2 * n + c_1) \text{ (Simetría)}$$

$$O(c_2 * n + c_1) = O(c_2 * n) \text{ (regla de la suma)}$$

$$O(c_2 * n) = O(n) \text{ (regla del producto)}$$

$T(n)$ es $O(n)$ es decir que la complejidad asintótica es lineal.

Donde n es el número del cual se busca hallar su factorial

- fibonacci

Complejidad para el peor de los casos:

$$T(n) = c_2 + T(n-1) + T(n-2), \text{ donde } c_2=4$$

ESTRUCTURA DE DATOS 1

Código ST0245

$$T(n) = c_2 * 2^n + c_1$$

$$T(n) = O(c_2 * 2^n + c_1) \text{ (Simetría)}$$

$$O(c_2 * 2^n + c_1) = O(c_2 * 2^n) \text{ (regla de la suma)}$$

$$O(c_2 * 2^n) = O(2^n) \text{ (regla del producto)}$$

$T(n)$ es $O(2^n)$ es decir que la complejidad asintótica es exponencial.

Donde n es el enésimo número de la serie fibonacci

Complejidad de ejercicios recursión 2

- **groupSum6:**

Complejidad para el peor de los casos:

$$t(n) = c_1 + t(n-1) + t(n-1), \text{ donde } c_1 = 7$$

$$t(n) = c_1 * (2^n - 1) + c_1 * 2^{(n-1)}$$

$$O(c_1 * (2^n - 1) + c_1 * 2^{(n-1)}) \text{ (Simetría)}$$

$$O((2^n - 1) + 2^{(n-1)}) \text{ (regla del producto)}$$

$$O(2^n + 2^n) \text{ (regla de la suma)}$$

$$O(2 * 2^n)$$

$$O(2^n) \text{ (regla del producto)}$$

complejidad: $O(2^n)$ exponencial

Donde n es el tamaño de la matriz de enteros

- **groupNoAdj:**

Complejidad para el peor de los casos:

$$t(n) = c_1 + t(n-2) + t(n-1), \text{ donde } c_1 = 7$$

ESTRUCTURA DE DATOS 1

Código ST0245

$$t(n) = 2^n$$

complejidad: $O(2^n)$ exponencial

Donde n es el tamaño de la matriz de enteros

- **split53:**

Complejidad para el peor de los casos:

$$t(n) = c_1 + t(n-1) + t(n-1) \text{ donde } c_1 = 9$$

$$t(n) = c_1(2^n - 1) + c_1 2^{n-1}$$

$$t(n) = (2^n - 1) + 2^{n-1} \text{ (regla del producto)}$$

$$t(n) = 2^n + 2^{n-1} \text{ (regla de la suma)}$$

$$t(n) = 2 * 2^{n-1}$$

$$t(n) = 2^n \text{ (regla del producto)}$$

complejidad: $O(2^n)$ exponencial

Donde n es el tamaño de la matriz de enteros

- **splitArray**

Complejidad para el peor de los casos:

$$T(n) = c_2 + T(n-1) + T(n-1), \text{ donde } c_2 = 8$$

$$T(n) = c_2(2^n - 1) + c_1 2^{n-1}$$

$$T(n) = O(c_2(2^n - 1) + c_1 2^{n-1}) \text{ (Simetría)}$$

$$O(c_2(2^n - 1) + c_1 2^{n-1}) = O(c_2(2^n - 1)) \text{ (regla de la suma)}$$

$$O(c_2(2^n - 1)) = O(2^n) \text{ (regla del producto)}$$

$$O(2^n) = O(2^n) \text{ (regla de la suma)}$$

ESTRUCTURA DE DATOS 1

Código ST0245

$T(n)$ es $O(2^n)$ es decir que la complejidad asintótica es exponencial.

Donde n es el tamaño del arreglo de enteros

- splitOdd10

Complejidad para el peor de los casos:

$$T(n) = c2 + T(n-1) + T(n-1), \text{ donde } c2 = 8$$

$$T(n) = c2 * (2^n - 1) + c1 * 2^{n-1}$$

$$T(n) = O(c2 * (2^n - 1) + c1 * 2^{n-1}) \text{ (Simetría)}$$

$$O(c2 * (2^n - 1) + c1 * 2^{n-1}) = O(c2 * (2^n - 1)) \text{ (regla de la suma)}$$

$$O(c2 * (2^n - 1)) = O(2^n - 1) \text{ (regla del producto)}$$

$$O(2^n - 1) = O(2^n) \text{ (regla de la suma)}$$

$T(n)$ es $O(2^n)$ es decir que la complejidad asintótica es exponencial.

Donde n es el tamaño del arreglo de enteros

- groupSum5

Complejidad para el peor de los casos:

$$T(n) = c4 + T(n-1) + T(n-1), \text{ donde } c4 = 8$$

$$T(n) = c4 * (2^n - 1) + c4 * 2^{n-1}$$

ESTRUCTURA DE DATOS 1

Código ST0245

$$T(n) = O(c_4 * (2^n - 1) + c_1 * 2^{n-1}) \text{ (Simetría)}$$

$$O(c_4 * (2^n - 1) + c_1 * 2^{n-1}) = O(c_4 * (2^n - 1)) \text{ (regla de la suma)}$$

$$O(c_4 * (2^n - 1)) = O(2^n - 1) \text{ (regla del producto)}$$

$$O(2^n - 1) = O(2^n) \text{ (regla de la suma)}$$

$T(n)$ es $O(2^n)$ es decir que la complejidad asintótica es exponencial.

Donde n es el tamaño del arreglo de enteros

4) Simulacro de Parcial

4.1

4.1.1. Completa la línea 6

a. s.substring(0, i)

4.1.2. Completa la línea 9

c. true

4.1.3. Completa la línea 11

a. solve(t, s.substring(i), n - i)

4.3 La complejidad de la función mystery es

b. $T(n,m) = C \times n \times m^2$

ESTRUCTURA DE DATOS 1

Código ST0245

4.4 Sucesión Lucas

```
1 int lucas(int n){  
2   if(n == 0) return 2;  
3   if(n == 1) return 1;  
4   return lucas(n-1) + lucas(n-2);  
5 }
```

4.4.1 La complejidad asintótica del algoritmo anterior, para el peor de los casos, en términos de n , es:

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

4.5

4.5.1. Completa la línea 3

a. true

4.5.2. Completa la línea 4

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

6) Trabajo en Equipo y Progreso Gradual

ESTRUCTURA DE DATOS 1

Código ST0245

6.1 Actas de reunión

Acta	Fecha	Integrante	Hecho
1	21/08/2020	Laura Katterien Zapata Rendón	Leer guía y laboratorio 1
		Maria Alejandra Velez Clavijo	Leer guía y laboratorio 1
2	22/08/2020	Laura Katterien Zapata Rendón	Conectar Netbeans con gitHub
			Implementar ejercicio 1.1
		Maria Alejandra Velez Clavijo	Conectar Netbeans con gitHub
			Implementar ejercicio 1.2
3	23/08/2020	Laura Katterien Zapata Rendón	Documentar ejercicio 1.1
			Hacer 3 ejercicios recursión 2
		Maria Alejandra Velez Clavijo	Documentar ejercicio 1.2
			Hacer 3 ejercicios recursión 2
4	24/08/2020	Laura Katterien Zapata Rendón	3.5 Calcular la complejidad de los ejercicios en línea
			3.1 Calcular complejidad asintótica 1.1
		Maria Alejandra Velez Clavijo	3.5 Calcular la complejidad de los ejercicios en línea
			3.2 Tomar tiempo para 20 tamaños diferentes en el ejercicio 1.1
5	25/08/2020	Laura Katterien Zapata Rendón	3.6 Explicar variables
			Simulacro parcial impares (4.1, 4.3 4.5)
			3.2 Generar gráfica y analizar resultados
		Maria Alejandra Velez Clavijo	3.6 Explicar variables
			Simulacro parcial pares (4.2, 4.4, 4.6)
			3.2 Estimar tiempo 300.000 caracteres en el ejercicio 1.1

ESTRUCTURA DE DATOS 1

Código ST0245

Acta	Haciendo	Por hacer
1	Conectar Netbeans con gitHub	Implementar ejercicio 1.1
		Hacer 3 ejercicios recursión 1
	Conectar Netbeans con gitHub	Implementar ejercicio 1.2
		Hacer 3 ejercicios recursión 1
2	Hacer 3 ejercicios recursión 1	Documentar ejercicio 1.1
		Hacer 3 ejercicios recursión 2
	Hacer 3 ejercicios recursión 1	Documentar ejercicio 1.2
		Hacer 3 ejercicios recursión 2
3	3.3 Responder si la complejidad del 1.1 es apropiada	3.1 Calcular complejidad asintótica 1.1
		3.2 Generar gráfica y analizar resultados
		3.5 Calcular la complejidad de los ejercicios en linea
	3.3 Responder si la complejidad del 1.1 es apropiada	3.2 Tomar tiempo para 20 tamaños diferentes en el ejercicio 1.1
		3.2 Estimar tiempo 300.000 caracteres en el ejercicio 1.1
		3.5 Calcular la complejidad de los ejercicios en linea
4	3.2 Generar gráfica y analizar resultados	3.6 Explicar variables
		Simulacro parcial impares (4.1, 4.3 4.5)
	3.2 Estimar tiempo 300.000 caracteres en el ejercicio 1.1	3.6 Explicar variables
		Simulacro parcial pares (4.2, 4.4, 4.6)
5		

ESTRUCTURA DE DATOS 1

Código ST0245

6.3 El reporte de cambios del informe de laboratorio

← Hoy, 22:28

100%

Total: 25 cambios

ESTRUCTURA DE DATOS 1

Código ST0245

Laboratorio Nro. X1

Escribir el tema del laboratorio

Rekursión

Laura Katterine Zapata Rendón

Universidad Eafit

Medellín, Colombia

lkzapatar@eafit.edu.co

Maria Alejandra Vélez Clavijo

Universidad Eafit

Medellín, Colombia

mavelezc1@eafit.edu.co

Historial de versiones

Mostrar solo versiones con nombre

HOY

25 de agosto, 22:28

Versión actual

Laura Katterine Zapata Rendón

Maria Alejandra

25 de agosto, 20:21

Maria Alejandra

Laura Katterine Zapata Rendón

25 de agosto, 10:02

Maria Alejandra

AYER

24 de agosto, 23:16

Maria Alejandra

24 de agosto, 22:06

Laura Katterine Zapata Rendón

24 de agosto, 21:01

Mostrar cambios

Historial de versiones

Mostrar solo versiones con nombre

HOY

25 de agosto, 22:28

Versión actual

Laura Katterine Zapata Rendón

Maria Alejandra

25 de agosto, 20:21

Maria Alejandra

Laura Katterine Zapata Rendón

25 de agosto, 10:02

Maria Alejandra

AYER

24 de agosto, 23:16

Maria Alejandra

24 de agosto, 22:06

Laura Katterine Zapata Rendón

24 de agosto, 21:01

Mostrar cambios

Historial de versiones

Mostrar solo versiones con nombre

Maria Alejandra

25 de agosto, 20:21

Maria Alejandra

Laura Katterine Zapata Rendón

25 de agosto, 10:02

Maria Alejandra

AYER

24 de agosto, 23:16

Maria Alejandra

24 de agosto, 22:06

Laura Katterine Zapata Rendón

24 de agosto, 21:01

Laura Katterine Zapata Rendón

24 de agosto, 20:56

Laura Katterine Zapata Rendón

Mostrar cambios

6.2 El reporte de cambios en el código



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

[kattezapata](#) / [ST0245-001](#)

- <> Code
- ! Issues
- Pull requests
- Actions
- Projects
- Wiki
- Security
- Insights

History for [ST0245-001](#) / [laboratorios](#) / [lab01](#)

Commits on Aug 25, 2020

Se actualiza el punto 1.1 en la clase <code>CodigoEjercicio1</code> kattezapata committed 1 hour ago
Se organizan ejercicios recursion 1 y recursion 2 kattezapata committed 2 hours ago
Complejidad asintótica punto 1.1 kattezapata committed 3 hours ago
Complejidad asintótica punto 1.1 kattezapata committed 4 hours ago
Complejidad asintótica punto 1.1 kattezapata committed 4 hours ago
Complejidad asintótica punto 1.1 kattezapata committed 5 hours ago
Complejidad asintotica ejercicios de recursividad 1 y 2 de codingBat kattezapata committed 8 hours ago
Complejidad asintotica ejercicios de recursividad 1 y 2 de codingBat kattezapata committed 8 hours ago
Documentacion de los ejercicios de recursividad 1 y 2 de codingBat

 kattezapata committed 13 hours ago

Se calcula la complejidad asintotica del ejercicio 1.1 : Subsecuencia... ...

 kattezapata committed 14 hours ago

Se calcula la complejidad asintotica del ejercicio 1.1 : Subsecuencia... ...

 kattezapata committed 14 hours ago

Se implementa el ejercicio 1.1 : Subsecuencia en comun mas larga entr... ...

 kattezapata committed 14 hours ago

Se implementan 3 ejercicios de recursividad 2 en codingBat : splitArr... ...

 kattezapata committed 22 hours ago

Commits on Aug 24, 2020

Documentación de ejercicios en línea

 malejav02 committed yesterday

Corrección del ejercicio 1.2 ...

 malejav02 committed yesterday

Se implementan 3 ejercicios de recursividad 1 en codingBat : fibonacc... ...

 kattezapata committed yesterday

Complejidad de los algoritmos

 malejav02 committed yesterday

Ejercicios recursión 1 (CodingBat) ...

 malejav02 committed yesterday

Commits on Aug 23, 2020

Ejercicios recursión 1 (CodingBat)

 malejav02 committed 2 days ago

Implementación ejercicio 1.2

 malejav02 committed 2 days ago

Implementación ejercicio 1.2

 malejav02 committed 2 days ago

cambio de versión en clase ejercicio 1

 malejav02 committed 2 days ago

Título claseCodigoEjercicio2

 kattezapata committed 2 days ago

Se añaden version de la clase CodigoEjercicio2

 kattezapata committed 2 days ago

Se añaden version de la clase CodigoEjercicio1

 **kattezapata** committed 2 days ago

Se añaden autores de la clase CodigoEjercicio1

 **LauraZapata** committed 2 days ago

Se añaden autores de la clase CodigoEjercicio2

 **LauraZapata** committed 2 days ago

Se añaden autores de la clase CodigoEjercicio2

 **LauraZapata** committed 2 days ago

Delete CodigoEjercicio1.java

 **kattezapata** committed 2 days ago

Add files via upload

 **kattezapata** committed 2 days ago

Add files via upload

 **kattezapata** committed 2 days ago

🔗 Commits on Feb 8, 2018

cambios al lab 1

 **mauriciotoro** committed on 8 Feb 2018

🔗 Commits on Nov 23, 2017

borro cosas que sobran de archivos

 **mauriciotoro** committed on 23 Nov 2017

🔗 Commits on Aug 1, 2017

Agregando estructura

 **jarcil13** committed on 1 Aug 2017

[Newer](#)

[Older](#)