

Laboratorio Nro. 1

Recursión y Complejidad Asintótica

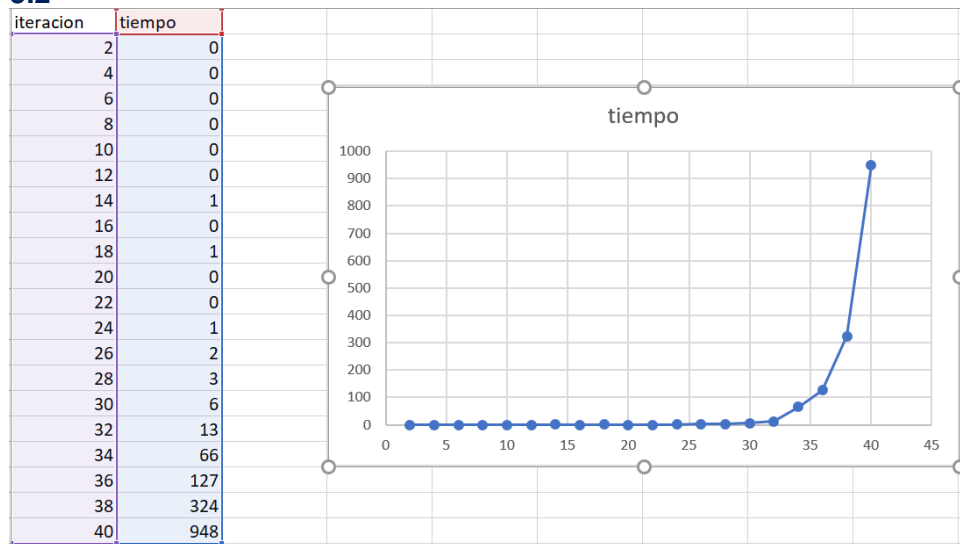
Stiven Yepes
Universidad Eafit
Medellín, Colombia
esyepesv@eafit.edu.co

Sara Rodríguez
Universidad Eafit
Medellín, Colombia
srodriguev@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

$$3.1 \quad T(n) = T(n-1) + T(n-2) + c$$

3.2



- Podemos ver que los datos siguen una tendencia exponencial, la cual aproximadamente a partir de $n=38$ tiene un cambio más brusco con respecto a los datos anteriores, lo que nos dice que a partir de ese punto el tiempo seguirá aumentando cada vez más con esta misma tendencia. Igualmente, podemos concluir por la naturaleza de la gráfica, que este es un algoritmo que puede ser de orden n^3 o n^2 .
- Corriendo el programa, el resultado para $n=50$ fue: 137148 ms.

3.3

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

No, la complejidad de este algoritmo hace casi imposible aplicarlo a números muy altos como los que se emplearían en Puerto Antioquia, ya que el tiempo de ejecución de estos sería exorbitante, poco eficiente e incluso imposible para máquinas normales.

3.5

Recursion-1:

- PowerN: $T(n) = T(n-1) + c$, donde n es la potencia a la que se va a elevar la base.
- CountX: $T(n) = T(n-1) + T(n-1) + c$, donde n es la longitud de la nueva cadena de caracteres(str.substring) creada a partir de la cadena original(str).
- ChangePi: $T(n) = T(n-2) + T(n-1) + c$, donde n es la longitud de la nueva cadena de caracteres(str.substring) creada a partir de la cadena original(str).
- AllStar: $T(n) = T(n-1) + c$, donde n es la longitud de la nueva cadena de caracteres(str.substring) creada a partir de la cadena original(str).
- EndX: $T(n) = T(n-2) + T(n-1) + c$, donde n es la longitud de la nueva cadena de caracteres(str.substring) creada a partir de la cadena original(str).

Recursion-2:

- GroupNoAdj: $T(n) = T(n-2) + T(n-1) + c$, donde n es lo que le falta a start para llegar a la longitud del arreglo y así terminar de recorrerlo.
- SplitArray: $T(n) = T(n-1) + T(n-1) + c$, donde n es la diferencia entre la cantidad de números en el arreglo(nums) y el número actual en el contador i .
- GroupSum6: $T(n) = T(n-1) + T(n-1) + T(n-1) + c$, donde n es lo que le falta a start para llegar a la longitud del arreglo y así terminar de recorrerlo.
- GroupSumClump: $T(n) = n + T(n-1) + T(n-1) + c$, donde n es lo que le falta a start para llegar a la longitud del arreglo y así terminar de recorrerlo.
- IsSideOdd: $T(n) = T(n-1) + T(n-1) + c$, donde n es lo que le falta a i para llegar a la longitud del arreglo y así terminar de recorrerlo.

4) Simulacro de Parcial

4.1

Línea 3: true;

Línea 4: `s.charAt(0) == s.charAt(s.length()-1)`

4.2

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

a) $T(n)=T(n/2)+C$

4.3

4.3.1 Línea 4: $n-a, a, b, c$

4.3.2 Línea 5: $\text{res}, \text{solucionar}(n-b, a, b, c)+1$

4.3.3 Línea 6: $\text{res}, \text{solucionar}(n-c, a, b, c)+1$

4.4 (Opc)

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

4.5

4.5.1

Línea 3: $\text{if}(T==0)$ return 1;

Línea 4: $\text{if}(T \leq 1)$ return 0;

Línea 8: return $f1+f2+f3$;

4.5.2

a. $T(n)=T(n-1)+C$

4.6

4.6.1. Línea 10: $\text{sumaAux}(n.\text{substring}(2+i), i++)$;

4.6.2. Línea 12: $\text{sumaAux}(n.\text{substring}(i+1), i++)$;

4.8

4.8.1 Línea 9: return 0;

4.8.2 Línea 13: $ni+nj$

4.9 (Opc)

c. 22

4.10

b. 6

4.11

4.11.1. Línea 4: $n - 1$, $\text{lucas}(n - 2)$;

4.11.2 $T(n)=T(n-1)+T(n-2)+c$, que es $O(2^n)$

4.12

4.12.1 Línea 13: sat

4.12.2 Línea 17: $\text{Math.max}(fi, fj)$

4.12.3 Línea 18: sat

5) Lectura recomendada (opcional)

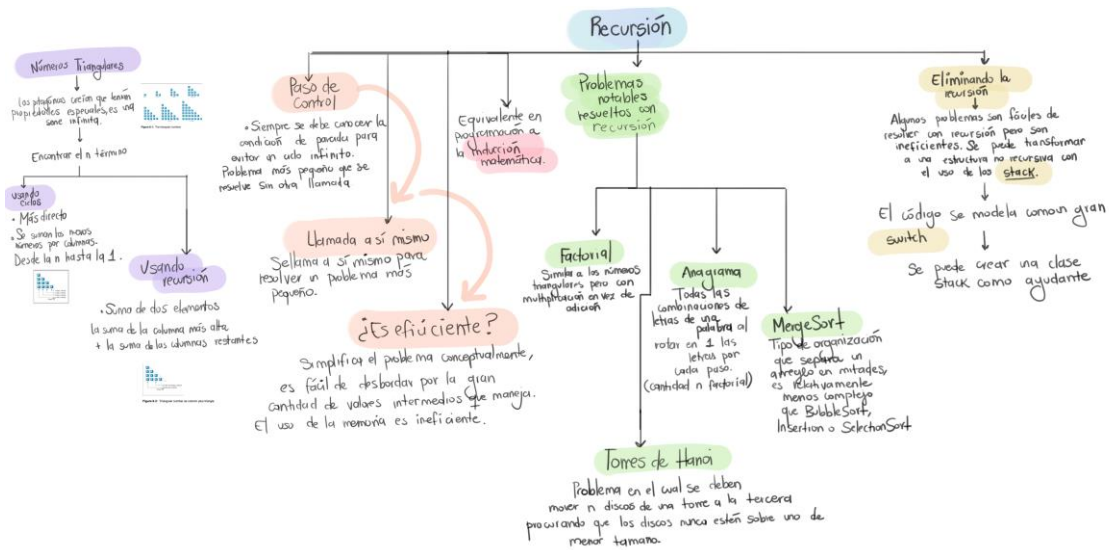
Mapa conceptual:

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



Disponible para visualización en tamaño real en el siguiente enlace:

[https://eafit-](https://eafit-my.sharepoint.com/:i/g/personal/srodriguev_eafit_edu_co/EeXyicE4ZlICm1SIKx6nSEEBBaIA5UhUpArPEldcZE5gzQ?e=N6Nx9K)

[my.sharepoint.com/:i/g/personal/srodriguev_eafit_edu_co/EeXyicE4ZlICm1SIKx6nSEEBBaIA5UhUpArPEldcZE5gzQ?e=N6Nx9K](https://eafit-my.sharepoint.com/:i/g/personal/srodriguev_eafit_edu_co/EeXyicE4ZlICm1SIKx6nSEEBBaIA5UhUpArPEldcZE5gzQ?e=N6Nx9K)

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