



Taller 4

Análisis de Algoritmos

Carlos Andres Delgado S, Msg
`cadelgado1@usbcali.edu.co`

Septiembre de 2021

Reglas del taller

1. El taller pueden ser realizado por grupos de hasta 4 personas
2. Entregue la solución de los puntos del taller en un archivo en formato PDF. Incluya los códigos desarrollados en el archivo.
3. Entregue el taller en el enlace dispuesto por el campus virtual.
4. Se debe hacer una entrega por grupo, en caso de tener varias entregas por grupo se valdrá únicamente la primera que el docente revise
5. El plazo para entregar el taller será Lunes 01 de Noviembre de 2021 a las 23:59:59 hora de Colombia, si entrega después de esta fecha recibirá una penalización

Enunciado del taller

1. (35 %) Análisis algoritmo Divide y vencerás.

El siguiente algoritmo de ordenamiento es conocido como **StoogeSort**.

```
STOOGESORT( $A, i, j$ )
1 if  $A[i] > A[j]$ 
2   then exchange  $A[i] \leftrightarrow A[j]$ 
3 if  $i + 1 \geq j$ 
4   then return
5  $k \leftarrow \lfloor (j - i + 1) / 3 \rfloor$       ▷ Round down.
6 STOOGESORT( $A, i, j - k$ )           ▷ First two-thirds.
7 STOOGESORT( $A, i + k, j$ )           ▷ Last two-thirds.
8 STOOGESORT( $A, i, j - k$ )           ▷ First two-thirds again.
```

- (5 %) Explique la estrategia: Dividir, conquistar y combinar
- (10 %) Implemente el algoritmo
- (5 %) Realice pruebas con arreglos aleatorios de tamaño 10, 100, 1000, 10000.
- (15 %) Calcule la complejidad teórica del algoritmo y compárela con la complejidad práctica encontrada. Analice lo que encuentra.

2. (35 %) Diseño algoritmo Divide y vencerás.

Diseñe un algoritmo bajo la estrategia divide y vencerás para encontrar la moda de un vector. La moda de un vector es el elemento que más se repite, si existe más de una moda, el algoritmo retornará todos los elementos que son moda. Ejemplo

- (1,2,2,3,4) La moda es 2
- (1,2,2,3,3,5) La moda es 2 y 3

- (5 %) Explique la estrategia: Dividir, conquistar y combinar
- (10 %) Implemente el algoritmo
- (5 %) Realice pruebas con arreglos aleatorios de tamaño 10, 100, 1000, 10000.
- (15 %) Calcule la complejidad teórica del algoritmo y compárela con la complejidad práctica encontrada. Analice lo que encuentra.

3. (30 %) Comparación ejecución algoritmos

Implemente los algoritmos Insertion-Sort y Merge-Sort y realice una comparación entre los utilizando una tabla para entradas aleatorias de tamaño 10, 50, 100 y 500

El análisis debe estar contenido en una tabla donde, para cada algoritmo, se tomen muestras de diferentes tamaños (unas 2 o 3 muestras por cada n , por lo que debe tomar el tiempo promedio), se de el tiempo real de ordenamiento, la complejidad temporal del algoritmo, y se saque el factor constante¹.

Un ejemplo de la tabla (para un algoritmo de complejidad $\mathcal{O}(n^2)$) es el siguiente:

¹Recuerde que $Tiempo\ Real = Constante * Complejidad$

Entrada (n)	Tiempo Real (seg.)	Complejidad ($\mathcal{O}(n^2)$)	Constantes
5	0.11	25	0.0044
5	0.11	25	0.0044
5	0.12	25	0.0048
10	0.38	100	0.0038
10	0.40	100	0.0040
10	0.41	100	0.0041
\vdots			
Constante:			0.0042