

Laboratorio Nro. 2: Notación O grande

Juliana Henao Arroyave

Universidad Eafit
Medellín, Colombia
jhenaoa4@eafit.edu.co

Gerónimo Zuluaga Londoño

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

3) Simulacro de preguntas de sustentación de Proyectos

1. Esta en otro pdf.
2. Está en otro pdf.
3. Teniendo en cuenta lo anterior, el merge sort es mas eficiente para arreglos de tamaños muy grandes que el insertion sort, debido a que el merge sort divide el arreglo, es decir que simplifica un poco el problema, mientras que el insertion sort tiene que recorrer todo el arreglo uno a uno, haciéndolo asi menos eficiente a la hora de evaluar arreglos con una gran cantidad de elementos, tales como millones de datos.
4. El algoritmo empleado para maxSpan consiste en que se tiene un arreglo y un ciclo anidado dentro de otro. El ciclo de afuera indica la posición del número que se va a buscar su repetido por medio del ciclo interior, que va de atrás hacia adelante y solo llega hasta donde se encuentra el primer ciclo (porque de ahí para atrás no va a encontrar otro número igual), esto se debe a que hay que buscar el número más a la derecha de ese valor. Luego se calcula cuantos números hay entre el de más a la izquierda y el de más a la derecha, incluyéndolos a ambos; esto se hace restando la posición del de la derecha con la de la izquierda y sumado uno para contar a ambos. Pero como puede haber varios números repetidos en el arreglo, se toman los que estén más separados. Para saber esto es necesario crear una variable que sea el máximo de todos y que, si hay algún intervalo mayor que este máximo, este intervalo será el nuevo máximo. Cuando se termine de recorrer el ciclo se retorna la variable con el intervalo máximo.

5. Array 2

sum13:

$$T(n) = n + C$$

$$T(n) = n$$

$$T(n) = O(n)$$

centeredAverage:

$$T(n) = n + n + C$$

$$T(n) = 2n$$

$$T(n) = O(n)$$

countEvens:

$$T(n) = n + C$$

$$T(n) = n$$

$$T(n) = O(n)$$

bigDiff:

$$T(n) = n + C$$

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

$$T(n) = n$$

$$T(n) = O(n)$$

lucky13:

$$T(n) = n + C$$

$$T(n) = n$$

$$T(n) = O(n)$$

Array 3

maxSpan:

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

$$T(n) = n + n^2 - n + C$$

$$T(n) = O(n^2)$$

fix34:

$$T(n) = n + n(n) + C$$

$$T(n) = n^2$$

$$T(n) = O(n^2)$$

fix45

$$T(n) = n + n(n) + C$$

$$T(n) = n^2$$

$$T(n) = O(n^2)$$

canBalance

$$T(n) = n * n + C$$

$$T(n) = n * n$$

$$T(n) = O(n^2)$$

MaxMirror

$$T(n) = n * n + C$$

$$T(n) = n * n$$

$$T(n) = O(n^2)$$

6. En la medida de la complejidad n es la variable en que esta la función, que indica cuantos pasos hace el programa (cíclico en ese caso, por ejemplo) o en otras palabras cuantas veces se ejecuta ciertas funciones del programa. En ciertos casos se usa otra letra como la m para distinguir que es un unmero diferente de pasos a realizar.

4) Simulacro de Parcial

1. c
2. b
3. b
4. b
5. d
6. d
7. 7.1 $T(n) = T(n-1)$
7.2 $O(n)$
8. b
9. d

10. *c*

11. *c*

12. *b*

13. *a*

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co