Universidad Nacional de Cuyo
Facultad de Ingeniería
Licenciatura en Ciencias de la Computación

Trabajo Práctico Nº 1

Algoritmos y Estructuras de Datos

Análisis de Complejidad

2024

Gonzalo Padilla Lumelli

Abril 2024

Ejercicio 1

Demuestre que $6n^3 \neq O(n^2)$.

Solución

$6n^3 = O(n^2)$ es, por definición, equivalente a decir que existen valores c y n_0 tal que $6n^3 < cn^2$ para $\forall n > n_0$. Se quiere demostrar lo opuesto, es decir, que para todo c y n_0, existe $n > n_0$ tal que $6n^3 \geq cn^2$. Analizamos los puntos en los que $6n^3$ y cn^2 coinciden, y los intervalos definidos por tales puntos:

$$\begin{aligned} 6n^3 &= cn^2 \\ 6n^3 - cn^2 &= 0 \\ n^2\left(n - \frac{c}{6}\right) &= 0 \end{aligned}$$

Puede verse que ambas funciones son iguales en $n = 0$ y $n = c/6$. Para $n > c/6$, la diferencia es no negativa y, por lo tanto, $6n^3 \geq cn^2$. Si $n_0 \leq c/6$, entonces basta tomar $n = \lfloor c/6 \rfloor + 1 \geq n_0$, y se tendrá que $n > n_0$, y $6n^3 \geq cn^2$. Si, en cambio, $n_0 > c/6$, basta tomar $n = n_0 + 1 > c/6$, y se tendrá también que $n > n_0$, y $6n^3 > cn^2$.

Entonces, $\forall c \forall n_0 \exists n : [(n > n_0) \wedge (6n^3 \geq cn^2)]$. Por lo tanto, $6n^3 \neq O(n^2)$.

Ejercicio 2

¿Cómo sería un array de números (mínimo 10 elementos) para el mejor caso de la estrategia de ordenación Quicksort(n)?

Solución

Ejercicio 3

Cuál es el tiempo de ejecución de la estrategia Quicksort(A), Insertion-Sort(A) y Merge-Sort(A) cuando todos los elementos del array A tienen el mismo valor?

Solución

Ejercicio 4

Implementar un algoritmo que ordene una lista de elementos donde siempre el elemento del medio de la lista contiene antes que él en la lista la mitad de los elementos menores que él. Explique la estrategia de ordenación utilizada.

Solución

Ejercicio 5

Implementar un algoritmo Contiene-Suma(A,n) que recibe una lista de enteros A y un entero n y devuelve True si existen en A un par de elementos que sumados den n. Analice el costo computacional.

Solución

Ejercicio 6

Investigar otro algoritmo de ordenamiento como BucketSort, HeapSort o RadixSort, brindando un ejemplo que explique su funcionamiento en un caso promedio. Mencionar su orden y explicar sus casos promedio, mejor y peor.

Solución

Ejercicio 7

A partir de las siguientes ecuaciones de recurrencia, encontrar la complejidad expresada en $\Theta(n)$ y ordenarlas de forma ascendente respecto a la velocidad de crecimiento. Asumiendo que T(n) es constante para $n \leq 2$. Resolver 3 de ellas con el método maestro completo: $T(n) = aT(n/b) + f(n)$ y otros 3 con el método maestro simplificado: $T(n) = aT(n/b) + n^c$

a. $T(n) = 2T(n/2) + n^4$
b. $T(n) = 2T(7n/10) + n$
c. $T(n) = 16T(n/4) + n^2$
d. $T(n) = 7T(n/3) + n^2$
e. $T(n) = 7T(n/2) + n^2$
f. $T(n) = 2T(n/4) + \sqrt{n}$

Solución