◁ Ejercicio 1 ▷

Ordene el vector $[9, 1, 3, 5, 0, 4, 2, 6, 8, 7]$ utilizando los algoritmos mergesort y quicksort.

◁ Ejercicio 2 ▷

Suponga que existen tres algoritmos diferentes, A, B y C, para resolver un determinado problema:

- El algoritmo A resuelve la instancia del problema de tamaño n dividiéndola en dos subinstancias de tamaño $n/2$, resuelve recursivamente cada subinstancia y combina las soluciones en tiempo lineal.
- El algoritmo B resuelve la instancia del problema de tamaño n resolviendo dos subinstancias de tamaño $n-1$ y combinando las soluciones en tiempo constante.
- El algoritmo C resuelve la instancia del problema de tamaño n dividiéndola en 9 subinstancias de tamaño $n/3$, resuelve recursivamente cada una de estas subinstancias y combina las soluciones en tiempo $\mathsf{O}(n^3)$.

¿Cuál es el orden de eficiencia de cada uno de estos algoritmos? ¿Cuál escogería para resolver el problema?

◁ Ejercicio 3 ▷

Se desea conocer si, dado un número natural n, este es o no un número combinatorio del tipo $\binom{k}{2}$, para algún número k. Por ejemplo, $36 = \binom{9}{2}$, o $499500 = \binom{1000}{2}$. Diseñe un algoritmo lo más eficiente posible para resolver este problema. ¿Cuál es su orden de eficiencia? Nota: $\binom{k}{2} = \frac{k\cdot(k-1)}{2}$.

◁ Ejercicio 4 ▷

Un número natural n se dice que es cuadrado perfecto si se corresponde con el cuadrado de otro número natural. Por ejemplo, $4 = 2^2$, $25 = 5^2$, o $100 = 10^2$, son cuadrados perfectos. Diseñe un algoritmo «divide y vencerás» que, dado un número natural n, determine si el número es cuadrado perfecto o no, con un orden de eficiencia lo mejor posible. ¿Cuál es su orden de eficiencia? Nota: no es posible emplear la operación raíz cuadrada.

◁ Ejercicio 5 ▷

Dado un vector v que contiene una secuencia de n enteros, se pretende localizar subsecuencias consecutivas de valores $x_i, \ldots, x_j$ que cumplan la propiedad de regularidad siguiente: $|x_k - x_{k+1}| = M$, $\forall k$, $i \leq k < j$, para algún valor de M. Por ejemplo, dada la secuencia siguiente:

$$[0, 1, 3, 5, 3, 5, 0, 10, 9, 10, 11, 12, 11, 0, 0]$$

podemos encontrar, entre otras, las siguientes subsecuencias:

- $[0, 1]$ es una subsecuencia de tamaño 2 en la que $M = 1$.
- $[1, 3, 5, 3, 5]$ es una subsecuencia de tamaño 5 en la que $M = 2$.
- $[10, 9, 10, 11, 12, 11]$ es la subsecuencia más larga, de tamaño 6 y en la que $M = 1$.

Diseñe un algoritmo «divide y vencerás» para determinar la longitud de la subsecuencia consecutiva de longitud máxima que cumpla la propiedad anterior. ¿Cuál es el orden de eficiencia del algoritmo? Trate de diseñar un algoritmo cuyo orden de eficiencia sea mejor.

◁ Ejercicio 6 ▷

Dado un vector v de n números ordenados en orden creciente y donde los números no se repiten, y dado un rango numérico $[A, B]$, se desea saber cuántos elementos del vector están dentro de ese rango (cumplen que $A \leq x \leq B$). Diseñe un algoritmo lo más eficiente posible para resolver esta tarea. ¿Cuál es su orden de eficiencia?

◁ Ejercicio 7 ▷

Diseñe un algoritmo «divide y vencerás» para encontrar el par de puntos más cercano dentro de un conjunto de puntos en el plano. ¿Cuál es su orden de eficiencia?

◁ Ejercicio 8 ▷

Diseñe un algoritmo «divide y vencerás» para encontrar la moda de un vector de números enteros, es decir, el número entero que se repite más veces. ¿Cuál es su orden de eficiencia?

◁ Ejercicio 9 ▷

Dado un vector v ordenado (de forma no decreciente) de n números enteros, todos distintos, diseñe un algoritmo «divide y vencerás» que permita determinar si existe un índice i tal que $v[i] = i$ y encontrarlo en ese caso. ¿Cuál es el orden de eficiencia del algoritmo? Suponga ahora que los enteros no tienen por qué ser todos distintos (pueden repetirse). Determine si el algoritmo anterior sigue siendo válido, y, en caso negativo, diseñe uno que sí lo sea.

◁ Ejercicio 10 ▷

Se tienen k vectores ordenados (de menor a mayor), cada uno con n elementos. Se quiere combinarlos en un único vector ordenado (con $k \cdot n$ elementos). Una posible alternativa consiste en, usando un algoritmo clásico, mezclar los dos primeros vectores, posteriormente mezclar el resultado con el tercero, y así sucesivamente.

- ¿Cuál sería el tiempo de ejecución de este algoritmo?
- Diseñe un algoritmo de mezcla más eficiente.

◁ Ejercicio 11 ▷

Dados n enteros cualesquiera $a_1, a_2, \ldots, a_n$, se quiere calcular el valor de la expresión:

$$\max_{1 \leq i \leq j \leq n} \sum_{k=i}^{j} a_k$$

que calcula el máximo de las sumas parciales de elementos consecutivos. Por ejemplo, dados 6 números enteros $(-2, 11, -4, 13, -5, -2)$, la solución al problema es 20 (suma de a_2 hasta a_4). Diseñe un algoritmo de complejidad $\mathrm{O}(n \cdot \log(n))$ para resolver este problema. Ejemplifique el funcionamiento del algoritmo con la secuencia anterior.

◁ Ejercicio 12 ▷

Dado un vector v de n números enteros, diseñe un algoritmo de complejidad $\mathrm{O}(n \cdot \log(n))$ que determine si existen en el vector dos números enteros cuya suma sea X.