

Algoritmos y Estructuras de Datos

UT9-PD3

Santiago Blanco

29-06-2025

Ejercicio 1

Quicksort es un algoritmo bondadoso. Durante la ejecución de Quicksort, se cumple en todo momento que:

- Todo elemento ubicado a la izquierda del pivot es menor al pivote, y todo elemento a su derecha es mayor o igual.

Esta condición se mantiene luego de cada partición.

Ejercicio 2

INSERTE FOTO DE CUADERNO

Ejercicio 3

Dual-pivot

- Una variante que utiliza dos pivotes en lugar de un solo pivote, lo que permite dividir el array en tres partes simultáneamente.
- Introducida por Yaroslavskiy en 2009 y luego estandarizada en Java 7 para `Arrays.sort()`

Orden de tiempo de ejecución

Más eficiente en promedio.

- Promedio: $O(n \log n)$
- Peor caso: $O(n^2)$, aunque es muy poco frecuente por la elección inteligente de pivot
- En casos con muchas claves iguales, el de tres vías (3-way) puede alcanzar $O(n)$, aunque Java usa dual-pivot con optimizaciones heredadas de la técnica 3-way

Ejercicio 4

Ordena un arreglo dividiéndolo en dos partes alrededor de un pivote, de modo que todos los elementos menores que el pivote queden a su izquierda y los mayores o iguales a su derecha.

- Luego aplica recursivamente el mismo proceso a cada mitad.
- Es un algoritmo de tipo “divide y vencerás” (Divide and conquer).
- La función de partición es de orden lineal $O(n)$: recorre el arreglo una sola vez para ubicar correctamente los elementos respecto al pivote.
- Complejidad temporal del algoritmo:
 - Mejor caso: ocurre cuando el pivoto divide el arreglo en dos mitades iguales
 - Tiempo total: $O(n \log n)$
 - Peor caso: el pivot siempre es el menor o el mayor (por ejemplo, en un array ordenado)
 - Tiempo total: $O(n^2)$
 - Caso promedio: $O(n \log n)$

- Optimización: cuando el subarray es pequeño (por ejemplo, con menos de 10 elementos), se puede usar **Insertion Sort**, ya que es muy útil para arreglos casi ordenados o pequeños (Cercano a $O(n)$).