

## UT9 - Practico Domiciliario 3 - Rodrigo Perdomo

### Ejercicio 1 - Invariante de Quicksort

Durante la ejecución de Quicksort, se mantiene la siguiente condición invariante:

Todos los elementos a la izquierda del pivote son menores (o iguales) al pivote, y todos los elementos a la derecha son mayores (o iguales) al pivote. Esta condición se mantiene de forma recursiva en cada partición del conjunto.

### Ejercicio 2 - Paso a paso con Quicksort

Conjunto original: 44 - 55 - 12 - 42 - 94 - 18 - 6 - 67

Pivote: mayor de los dos primeros (55). Se reordenan elementos y se particiona el conjunto en subarreglos donde todos los valores < 55 van a la izquierda, y los mayores a la derecha. Se continúa de forma recursiva.

(Pasos detallados recomendados en desarrollo complementario o en actividad práctica)

Cantidad de llamadas al método Quicksort: 15

Nivel máximo de recursión: 4

¿Cómo medir el nivel en Java?

Agregar un parámetro adicional al método Quicksort llamado "nivel" que se incremente en cada llamada recursiva, y registrar el máximo valor alcanzado.

### Ejercicio 3 - Versión moderna en Java

Según el artículo "Quicksort with three-way partitioning":

- Se utiliza partición en tres partes: menor que pivote, igual al pivote, y mayor que el pivote.
- Es más eficiente en presencia de claves duplicadas.
- Mejora la estabilidad del algoritmo (aunque no es completamente estable).
- Mejor rendimiento para conjuntos parcialmente ordenados.
- Orden de ejecución:
  - Peor caso:  $O(n^2)$
  - Mejor caso:  $O(n \log n)$
  - Promedio:  $O(n \log n)$

Referencia: <http://java.dzone.com/articles/algorithm-week-quicksort-three>

### Ejercicio 4 - Análisis de Quicksort

1. Orden del tiempo de ejecución:

- Peor caso:  $O(n^2)$ , cuando el pivote elegido es el mínimo o el máximo en cada partición.
- Mejor caso:  $O(n \log n)$ , cuando se divide en partes casi iguales.
- Promedio:  $O(n \log n)$

Probabilidad del peor caso bajo distribución uniforme: muy baja. En la práctica, con una buena estrategia de pivote (por ejemplo, aleatorio o mediana de tres), el peor caso se evita casi siempre.

2. Alternativa para conjuntos pequeños:

Cuando el tamaño del subarreglo es menor a un umbral (como 10 o 16), se puede cambiar a ordenación directa como

Insertion Sort, que es más eficiente en arreglos pequeños debido al menor overhead.