

Profesor: Guillermo Palma

Estudiantes:

Haydeé Castillo Borgo. Carnet: 16-10209

Jesús Prieto. Carnet: 19-10211

Trimestre Abril-Julio 2023

## Laboratorio de la semana 4

En el presente laboratorio se llevó a cabo un estudio experimental de los algoritmos Quicksort, desarrollado en [1], Quicksort Three Way, desarrollado en [2], y Quicksort Dual Pivot, desarrollado en [3]; a través de un computador Intel® Core™ i5-2450M CPU @ 2.50GHz  $\times$  4, con 8Gb de RAM y sistema operativo Ubuntu 20.04.6 LTS. Para dicho estudio se empleó el lenguaje de programación Kotlin en su versión 1.8.21 y Java Virtual Machine JVM, versión 11.0.19.

Los tres algoritmos mencionados anteriormente son diferentes implementaciones de Quicksort, un algoritmo de ordenamiento basado en el esquema de Divide-and-Conquer; y el objetivo del presente estudio es verificar si las versiones mejoradas Three Way y Dual Pivot, en efecto son más eficientes que el Quicksort clásico.

Para ello se ejecutaron las tres versiones del algoritmo sobre un conjunto de arreglos generados aleatoriamente, de la siguiente manera:

Se consideraron cuatro tamaños:  $N = 500,000$ ,  $N = 1,000,000$ ,  $N = 1,500,000$  y  $N = 2,000,000$ ; y para cada  $N$  se generaron 10 arreglos, cuyos elementos venían dados por valores en el intervalo  $[0, N]$ . Luego, se ejecutó cada variante de Quicksort en cada uno de los 10 arreglos, se tomó el tiempo correspondiente a cada prueba y se calcularon los promedios y desviaciones estándar, agrupando las pruebas de acuerdo al tamaño del arreglo.

Dichos resultados se presentan en la siguiente tabla y en el siguiente gráfico.

| Algoritmo            | Tamaño de los arreglos<br>$N$ | Tiempo promedio<br>(en segundos) | Desviación estándar<br>(en segundos) |
|----------------------|-------------------------------|----------------------------------|--------------------------------------|
| Quicksort Clásico    | 500000                        | 0,2323                           | 0,0352                               |
| Quicksort Three Way  | 500000                        | 0,1652                           | 0,0492                               |
| Quicksort Dual Pivot | 500000                        | 0,2078                           | 0,0993                               |
| Quicksort Clásico    | 1000000                       | 0,4989                           | 0,0904                               |
| Quicksort Three Way  | 1000000                       | 0,3688                           | 0,1006                               |
| Quicksort Dual Pivot | 1000000                       | 0,4143                           | 0,0517                               |
| Quicksort Clásico    | 1500000                       | 0,9236                           | 0,0771                               |
| Quicksort Three Way  | 1500000                       | 0,6256                           | 0,1004                               |
| Quicksort Dual Pivot | 1500000                       | 0,7525                           | 0,0923                               |
| Quicksort Clásico    | 2000000                       | 1,299                            | 0,0908                               |
| Quicksort Three Way  | 2000000                       | 0,8963                           | 0,1081                               |
| Quicksort Dual Pivot | 2000000                       | 1,1885                           | 0,1898                               |

De lo obtenido se concluye que, en efecto, las variantes Three Way y Dual Pivot son más eficientes que el Quicksort clásico. Aún más, se obtuvo que la variante Three Way resultó ser la más eficiente de las tres.

## Referencias

- [1] Cormen, T., Leirserson, C., Rivest, R., and Stein, C. *Introduction to Algorithms*, 3ra ed. McGraw Hill, 2009.
- [2] Sedgewick, R., and Bentley, J. Quicksort is optimal. <https://sedgewick.io/wp-content/uploads/2022/03/2002QuicksortIsOptimal.pdf>, 2002. KnuthFest, Stanford University.

- [3] Wild, S., and Nebel, M. E. Average case analysis of java 7's dual pivot quicksort. In *Algorithms-ESA 2012: 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings 20* (2012), Springer, pp. 825–836.