Comparativa Quicksort

*Nota: Los subtítulos no son capturados en Xplore y no deben usarse

Alexis Raciel Ibarra Garnica
Facultad de Informática
Universidad Autómona de Querétaro
Santiago de Querétaro, Qro, México
direccion.correo@email.com

Pablo Natera Bravo
Facultad de Informática
Universidad Autómona de Querétaro
Santiago de Querétaro, Qro, México
pablonatera16@gmail.com

Abstract—In this paper we implemented QuickSort in C along with BubbleSort, Flag BubbleSort, SelectionSort and InsertionSort.

Index Terms—QuickSort, Divide and Conquer, Big O

I. Introducción

El algoritmo QuickSort, fue desarrollado por Tony Hoare en 1959 mientras era estudiante de ciencias de la computación en la Universidad Estatal de Moscú, surgió de la necesidad de ordenar listas de palabras para traducirlas del ruso al inglés. Su propuesta superó en eficiencia al método de Insertion Sort, introduciendo el concepto de particiones e implementándose inicialmente en Mercury Autocode.

Posteriormente, al regresar a Inglaterra, Hoare adaptó su idea para mejorar el algoritmo de Shellsort, lo que lo llevó a publicar QuickSort en 1961, y un año más tarde una versión mejorada. QuickSort está basado en el paradigma de Divide and Conquer, creando particiones recursivas en el arreglo hasta llegar a un caso base.

Los pasos principales que sigue el algoritmo son:

1) Si la longitud del arreglo es menor a dos, se retorna el elemento directamente (caso base).
2) Si la longitud es dos o más, se selecciona un pivote (existen distintas técnicas para ello).
3) El arreglo se reordena colocando los elementos menores al pivote a la izquierda y los mayores a la derecha.
4) Los pasos anteriores se aplican recursivamente a cada subarreglo hasta alcanzar el caso base.

En cuanto a la complejidad, el peor caso ocurre cuando las particiones dividen el arreglo de manera muy desigual (por ejemplo, $n-1$ y 0 elementos), resultando en una complejidad de $O(n^2)$. El mejor caso se logra cuando el pivote divide el arreglo en dos mitades balanceadas en cada paso, alcanzando una complejidad de $\Theta(n \log n)$. En la práctica, el caso promedio se acerca bastante al mejor caso debido a que la recurrencia converge a $n \log n$, por lo que QuickSort es considerado uno de los algoritmos de ordenamiento más eficientes.

Identifique la agencia de financiación aplicable aquí. Si no hay, elimine esta línea.

II. Metodología

Aquí debe describir los métodos, enfoques o procedimientos utilizados en su investigación.

III. Resultados

Presente los hallazgos y datos recopilados de su metodología. Incluya figuras y tablas como se muestra a continuación.

A. Subsección de Resultados (Ejemplo)

Si es necesario, puede subdividir sus secciones.

IV. Conclusiones

Resuma las conclusiones clave extraídas de los resultados y discuta las implicaciones de su trabajo.

Agradecimiento

El texto de agradecimiento va aquí (opcional).

Referencias

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

Fig. 1. Ejemplo de pie de figura sin archivo externo.

TABLE I
Ejemplo de Tabla.

Columna 1	Columna 2
Dato 1	Dato 2

Las plantillas de conferencia de la IEEE contienen texto guía para componer y formatear artículos de conferencia. Asegúrese de eliminar todo el texto de la plantilla de su artículo antes de enviarlo a la conferencia. Si no elimina el texto de la plantilla, su artículo podría no ser publicado.