



INSTITUTO POLITÉCNICO NACIONAL
Unidad Profesional Interdisciplinaria de Ingeniería
Campus Zacatecas.

MATERIA:

- **Án**álisis y Diseño de Algoritmos

Reporte 01:

- QuickSort

DOCENTE:

- Erika Sánchez Femat

Alumno:

- Luis Eduardo Espino Gutierrez

Introduccion

El método Quicksort es uno de los algoritmos de ordenación más icónicos y eficientes en el campo de la informática y las ciencias de la computación. Su enfoque divide y conquista, junto con su capacidad para realizar clasificaciones rápidas y efectivas, lo convierte en una herramienta fundamental para organizar conjuntos de datos de manera eficiente.

En este algoritmo, se elige un elemento como "pivote" y se reorganizan los datos de manera que los elementos más pequeños que el pivote estén a la izquierda, y los elementos más grandes estén a la derecha. Luego, este proceso se repite de manera recursiva en las sublistas generadas a ambos lados del pivote hasta que todo el conjunto de datos esté ordenado.

Quicksort se destaca por su eficiencia y velocidad en comparación con otros métodos de ordenación, con un rendimiento promedio de $O(n \log n)$ en tiempo de ejecución. Esto lo convierte en una herramienta valiosa en la programación y la gestión de datos, especialmente cuando se trata de grandes conjuntos de información.

En esta introducción, exploraremos en detalle cómo funciona Quicksort y por qué es tan ampliamente utilizado en el mundo de la informática y la tecnología.

Desarrollo

”QUICKSORT”

QUE ES...

El método de ordenamiento QuickSort es actualmente el más eficiente y veloz de los métodos de ordenación interna. Este método es una mejora sustancial del método de intercambio directo y recibe el nombre de QuickSort por la velocidad con que ordena los elementos del arreglo. Es un algoritmo basado en la técnica de divide y vencerás, que permite, en promedio, ordenar “n” elementos en un tiempo proporcional a “n Log n”.

COMO FUNCIONA...

La idea central de este algoritmo consiste en lo siguiente: Se toma un elemento “x” de una posición cualquiera del arreglo. Se trata de ubicar a “x” en la posición correcta del arreglo, de tal forma que todos los elementos que se encuentran a su izquierda sean menores o iguales a “x” y todos los elementos que se encuentren a su derecha sean mayores o iguales a “x”. Se repiten los pasos anteriores pero ahora para los conjuntos de datos que se encuentran a la izquierda y a la derecha de la posición correcta de “x” en el arreglo. Reubicar los demás elementos de la lista a cada lado del pivote, de manera que a un lado queden todos los menores que él, y al otro los mayores. En este momento, el pivote ocupa exactamente el lugar que le corresponderá en la lista ordenada. Repetir este proceso de forma recursiva para cada sublista mientras éstas contengan más de un elemento. Una vez terminado este proceso todos los elementos estarán ordenados.

Como se puede suponer, la eficiencia del algoritmo depende de la posición en la que termine el pivote elegido.

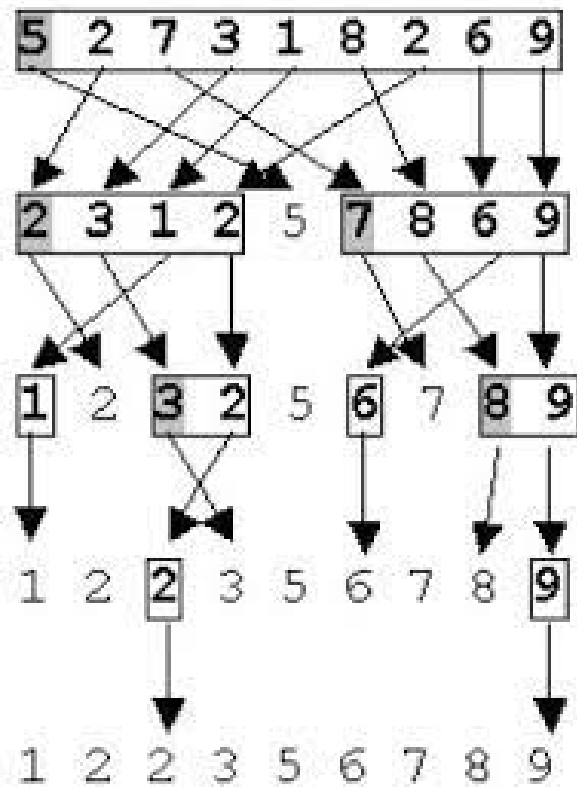


Figure 1: recorrido quicksort.

COMPLEJIDAD...

Quicksort tiene una complejidad de $O(n \log n)$ y es uno de los algoritmos mas utilizados para grandes volúmenes de datos, aunque no es el mas rapido.

MEJOR CASO...

La division inicial nos dara dos subvectores de igual tamaño, por lo tanto, la primera iteracion el vector complejo de tamaño n necesita $O(n)$.

La ordenacion de los dos sub-vectores restantes con $n/2$ elementos en la suma $2 \cdot O(n/2)$ para cada uno, por tanto $O(n \log n)$.

CASO PROMEDIO...

Es de igual forma la complejidad $O(\log n)$.

PEOR CASO...

El algoritmo selecciona solo un elemento en cada iteracion, por lo que $O(n) + O(n-1) + \dots + O(1)$, es igual a (n^2) .

Conclusiones

En conclusión, el método Quicksort es un algoritmo de ordenación extremadamente eficiente que presenta varias ventajas y algunas desventajas notables:

Ventajas:

Velocidad de Ejecución: Quicksort destaca por su rápido rendimiento promedio, lo que lo convierte en una elección excelente para ordenar conjuntos de datos grandes. Su tiempo de ejecución promedio es $O(n \log n)$, lo que lo hace muy eficiente en la práctica.

Dividir y conquistar: El enfoque dividir y conquistar permite una implementación elegante y eficiente del algoritmo, lo que lo hace fácil de entender y programar.

Desventajas:

Inestabilidad: Quicksort no garantiza la estabilidad en la clasificación. Esto significa que puede cambiar el orden relativo de elementos con claves iguales.

En resumen, Quicksort es un algoritmo de ordenación poderoso y eficiente que brinda un excelente rendimiento promedio y es especialmente útil para grandes conjuntos de datos. Sin embargo, su desempeño en el peor caso y su falta de estabilidad son aspectos a considerar al decidir si es la mejor opción para una tarea de ordenación específica. La elección de un pivote adecuado y una implementación cuidadosa son fundamentales para aprovechar al máximo las ventajas de Quicksort.

Referencias

<https://numerentur.org/quicksort/:text=Quicksort>

<https://es.theastrologypage.com/quicksort>

<https://tutospoo.jimdofree.com/tutoriales-java/m>