



Ventajas Ordenamientos Internos

Jhoel Chicaiza
Josué Merino
Daniel Vizcarra
PAblo Yanez



Ordenamientos Internos

La ordenación por intercambio consiste en comparar dos elementos del arreglo y determinar si existe un intercambio entre ellos, para esto debe definirse el tipo de ordenamiento que se quiere ya sea ascendente o descendente.

El ordenamiento interno consta de tres pasos en general:

- Leer los datos
- Ordenar los datos de memoria
- Escribir o presentar los datos

Procedimientos de ordenamiento interno:

- Burbuja
- Quicksort
- Shellsort
- Radix

Comparación Ordenamientos Internos

Burbuja	ShellSort	QuickSort	Radix
- Fácil implementación.	- Muy simple, tiempo de ejecución aceptable.	- Es muy rápida en su ejecución, no requiere memoria adicional.	- Es estable, preservando la orden de elementos iguales.
- No requiere memoria adicional.	- Trabajo muy bien con arreglos pequeños o medianos, no necesita de memoria externa.	- Tiene un lazo interior extremadamente corto.	- Funciona en un tiempo lineal, en comparación de varios otros métodos de ordenamiento.
- Es bastante sencillo, eficaz y en un código reducido se realiza mejor el ordenamiento.	- Eficiente para conjuntos de elementos medianos menores a 1000.	- Aprovecha la memoria caché del procesador del hardware, y eso por sí solo puede mejorar el rendimiento más de lo que se espera.	- El tiempo de ordenar cada elemento es constante, ya que no se hacen comparaciones entre elementos.



Burbuja

VENTAJAS	DESVENTAJAS
Popular y Fácil de implementar	No se comporta adecuadamente con una lista que contenga un número grande de elementos
Apropiado para la enseñanza académica	No es apropiado para aplicaciones y proyectos grandes, profesionales y complejos
Eficaz y sencillo	Consume bastante tiempo de computadora

QuickSort

VENTAJAS	DESVENTAJAS
Requiere de pocos recursos en comparación a otros métodos de ordenamiento.	Se complica la implementación si la recursión no es posible.
En la mayoría de los casos, se requiere aproximadamente $N \log N$ operaciones.	Peor caso, se requiere N^2
Ciclo interno es extremadamente corto.	Un simple error en la implementación puede pasar sin detección, lo que provocaría un rendimiento pésimo.
No se requiere de espacio adicional durante ejecución (in-place processing).	No es útil para aplicaciones de entrada dinámica, donde se requiere reordenar una lista de elementos con nuevos valores.
	Se pierde el orden relativo de elementos idénticos.

ShellSort

VENTAJAS	DESVENTAJAS
Es un algoritmo muy simple teniendo un tiempo de ejecución aceptable.	Su complejidad es difícil de calcular y depende mucho de la secuencia de incrementos que utilice.
No requiere memoria adicional.	Shell Sort es un algoritmo no estable porque se puede perder el orden relativo inicial con facilidad.
Fácil implementación.	Realiza numerosas comparaciones e intercambios.

Radix

VENTAJAS	DESVENTAJAS
Es estable, preservando la orden de elementos iguales.	No funciona tan bien cuando los números son muy largos, ya que el total de tiempo es proporcional a la longitud del número más grande y al número de elementos a ordenar.
Funciona en un tiempo lineal, en comparación de varios otros métodos de ordenamiento	
Es particularmente eficiente cuando se tratan con grandes grupos de números cortos.	



Bibliografía

- Solis, F. (2021). *Hacia el mundo de las Estructuras de Datos C++*.
- Wandy, J. (2017, 20 noviembre). *Las ventajas y desventajas de los algoritmos de ordenamiento*. Techlandia.
https://techlandia.com/ventajas-desventajas-algoritmos-ordenamiento-info_181515/
- Simpson, T. (2014, 21 mayo). *Radix Sort*. Radix Sort. Recuperado 15 de diciembre de 2021, de
<http://radixsort.byethost7.com/?i=1>
- 403 Forbidden. (2019). EDTeam. <https://ed.team/comunidad/ventajas-y-desventajas-del-metodo-de-ordenacion-por-burbuja>
- Sites.google.com. <https://sites.google.com/a/uabc.edu.mx/ordenacionshell/home/7---ventajas-y-desventajas>