

Métodos de Ordenamiento

Ordenamiento

Es la operación de arreglar los registros de una tabla en algún orden secuencial de acuerdo a un criterio de ordenamiento. El ordenamiento se efectúa con base en el valor de algún campo en un registro. El propósito principal de un ordenamiento es el de facilitar las búsquedas de los miembros del conjunto ordenado.

El ordenar un grupo de datos significa mover los datos o sus referencias para que queden en una secuencia tal que represente un orden, el cual puede ser numérico, alfabético o incluso alfanumérico, ascendente o descendente.

✓ **Métodos de Ordenamiento Elementales:**

1. Inserción
2. Selección
3. Burbujeo

✓ **Métodos de Ordenamiento no Elementales:**

1. Shell
2. Quick Sort
3. Fusión

❖ **Tipos de Ordenamiento:**

- Ordenamiento Interno → Ordenamiento de datos en Memoria Principal. (La lectura y grabación se hacen en registros)
- Ordenamiento Externo → Ordenamiento de datos en Disco.

❖ **Tipos de Entrada de Datos:**

- a. Entrada Ordenada = MEJOR CASO
- b. Entrada Orden Inverso = PEOR CASO
- c. Entrada Desordenada = CASO AL AZAR

❖ **Tipos de Algoritmo**

- Algoritmo Sensible: Modifica su tiempo de ejecución según el tipo de entrada.
- Algoritmo No Sensible: Su tiempo de ejecución es independiente al tipo de entrada.
- Algoritmo Estable: Aquellos que teniendo clave repetida, mantiene su posición inicial igual a la final.
- Algoritmo No Estable: Aquello que no respetan la posición inicial igual que la final teniendo claves repetidas

✓ **Métodos Elementales:**

1. ORDENAMIENTO POR SELECCIÓN

DESCRIPCIÓN.

- ✗ Buscas el elemento más pequeño de la lista.
- ✗ Lo intercambias con el elemento ubicado en la primera posición de la lista.
- ✗ Buscas el segundo elemento más pequeño de la lista.
- ✗ Lo intercambias con el elemento que ocupa la segunda posición en la lista.
- ✗ Repites este proceso hasta que hayas ordenado toda la lista.

ANÁLISIS DEL ALGORITMO.

- ✗ Requerimientos de Memoria: Al igual que el ordenamiento burbuja, este algoritmo sólo necesita una variable adicional para realizar los intercambios.
- ✗ Tiempo de Ejecución: El ciclo externo se ejecuta n veces para una lista de n elementos. Cada búsqueda requiere comparar todos los elementos no clasificados.

Ventajas:

1. Fácil implementación.
2. No requiere memoria adicional.
3. Rendimiento constante: poca diferencia entre el peor y el mejor caso.

Desventajas:

1. Lento.
2. Realiza numerosas comparaciones.

2. ORDENAMIENTO POR INSERCIÓN DIRECTA

DESCRIPCIÓN.

El algoritmo de ordenación por el método de inserción directa es un algoritmo relativamente sencillo y se comporta razonablemente bien en gran cantidad de situaciones.

Completa la triplete de los algoritmos de ordenación más básicos y de orden de complejidad cuadrático, junto con SelectionSort y BubbleSort.

Se basa en intentar construir una lista ordenada en el interior del array a ordenar.

De estos tres algoritmos es el que mejor resultado da a efectos prácticos. Realiza una cantidad de comparaciones bastante equilibrada con respecto a los intercambios, y tiene un par de características que lo hacen aventajar a los otros dos en la mayor parte de las situaciones.

Este algoritmo se basa en hacer comparaciones, así que para que realice su trabajo de ordenación son imprescindibles dos cosas: un array o estructura

similar de elementos comparables y un criterio claro de comparación, tal que dados dos elementos nos diga si están en orden o no.

En cada iteración del ciclo externo los elementos 0 a i forman una lista ordenada.

ANÁLISIS DEL ALGORITMO.

- Ø Estabilidad: Este algoritmo nunca intercambia registros con claves iguales. Por lo tanto es estable.
- Ø Requerimientos de Memoria: Una variable adicional para realizar los intercambios.
- Ø Tiempo de Ejecución: Para una lista de n elementos el ciclo externo se ejecuta n1 veces. El ciclo interno se ejecuta como máximo una vez en la primera iteración, 2 veces en la segunda, 3 veces en la tercera, etc.

Ventajas:

1. Fácil implementación.
2. Requerimientos mínimos de memoria.

Desventajas:

1. Lento.
2. Realiza numerosas comparaciones.

Este también es un algoritmo lento, pero puede ser de utilidad para listas que están ordenadas o semiordenadas, porque en ese caso realiza muy pocos desplazamientos.

3. METODO DE LA BURBUJA

DESCRIPCIÓN

La idea básica del ordenamiento de la burbuja es recorrer el conjunto de elementos en forma secuencial varias veces. Cada paso compara un elemento del conjunto con su sucesor ($x[i]$ con $x[i+1]$), e intercambia los dos elementos si no están en el orden adecuado.

El algoritmo utiliza una bandera que cambia cuando se realiza algún intercambio de valores, y permanece intacta cuando no se intercambia ningún valor, pudiendo así detener el ciclo y terminar el proceso de ordenamiento cuando no se realicen intercambios, lo que indica que este ya está ordenado.

Este algoritmo es de fácil comprensión y programación pero es poco eficiente puesto que existen $n-1$ pasos y $n-i$ comprobaciones en cada paso, aunque es mejor que el algoritmo de ordenamiento por intercambio.

En el peor de los casos cuando los elementos están en el orden inverso, el número máximo de recorridos es $n-1$ y el número de intercambios o comparaciones está dado por $(n-1) * (n-1) = n^2 - 2n + 1$. En el mejor de los casos cuando los elementos están en su orden, el número de recorridos es mínimo 1 y el ciclo de comparaciones es $n-1$.

La complejidad del algoritmo de la burbuja es $O(n)$ en el mejor de los casos y $O(n^2)$ en el peor de los casos, siendo su complejidad total $O(n^2)$.

✓ **Métodos No Elementales:**

4. ORDENAMIENTO POR EL MÉTODO DE SHELL

DESCRIPCIÓN

El método Shell es una versión mejorada del método de inserción directa. Este método también se conoce con el nombre de inserción con incrementos crecientes. En el método de ordenación por inserción directa cada elemento se compara para su ubicación correcta en el arreglo, con los elementos que se encuentran en la parte izquierda del mismo. Si el elemento a insertar es más pequeño que el grupo de elementos que se encuentran a su izquierda, es necesario efectuar entonces varias comparaciones antes de su ubicación.

Shell propone que las comparaciones entre elementos se efectúen con saltos de mayor tamaño pero con incrementos decrecientes, así, los elementos quedarán ordenados en el arreglo más rápidamente.

El Shell sort es una generalización del ordenamiento por inserción, teniendo en cuenta dos observaciones:

1. El ordenamiento por inserción es eficiente si la entrada está "casi ordenada".
2. El ordenamiento por inserción es ineficiente, en general, porque mueve los valores sólo una posición cada vez.

El algoritmo Shell sort mejora el ordenamiento por inserción comparando elementos separados por un espacio de varias posiciones. Esto permite que un elemento haga "pasos más grandes" hacia su posición esperada. Los pasos múltiples sobre los datos se hacen con tamaños de espacio cada vez más pequeños. El último paso del Shell sort es un simple ordenamiento por inserción, pero para entonces, ya está garantizado que los datos del vector están casi ordenados.

5. ORDENAMIENTO QUICK SORT

DESCRIPCIÓN

El ordenamiento por partición (Quick Sort) se puede definir en una forma más conveniente como un procedimiento recursivo.

Tiene aparentemente la propiedad de trabajar mejor para elementos de entrada desordenados completamente, que para elementos semiordenados. Esta situación es precisamente la opuesta al ordenamiento de burbuja.

Este tipo de algoritmos se basa en la técnica "divide y vencerás", o sea es más rápido y fácil ordenar dos arreglos o listas de datos pequeños, que un arreglo o lista grande.

Normalmente al inicio de la ordenación se escoge un elemento aproximadamente en la mitad del arreglo, así al empezar a ordenar, se debe

llegar a que el arreglo este ordenado respecto al punto de división o la mitad del arreglo.

Se podrá garantizar que los elementos a la izquierda de la mitad son los menores y los elementos a la derecha son los mayores.

Los siguientes pasos son llamados recursivos con el propósito de efectuar la ordenación por partición al arreglo izquierdo y al arreglo derecho, que se obtienen de la primera fase. El tamaño de esos arreglos en promedio se reduce a la mitad.

Así se continúa hasta que el tamaño de los arreglos a ordenar es 1, es decir, todos los elementos ya están ordenados.

En promedio para todos los elementos de entrada de tamaño n , el método hace $O(n \log n)$ comparaciones, el cual es relativamente eficiente.

6. ORDENAMIENTO POR MEZCLA

DESCRIPCIÓN

Mediante el enfoque de Dividir y conquistar, este algoritmo divide el arreglo inicial en dos arreglos donde cada uno contiene la mitad de los datos (partes iguales mas o menos uno), y se ordenan mediante sucesivos llamados recursivos para luego fusionar los resultados en el arreglo inicial.

Este algoritmo se basa en una función que permite mezclar dos vectores ordenados, produciendo como resultado un tercer vector ordenado que contiene los elementos de los dos vectores iniciales, el cual tiene una complejidad de $O(n)$ para mezclar dos arreglos, donde n es la suma de los tamaños de los dos arreglos.

El algoritmo principal que divide el arreglo, realiza $O(\log_2 n)$ particiones y como llama a la función que mezcla los dos arreglos y que tiene una complejidad de $O(n)$, entonces la complejidad total del algoritmo será de $O(n \log_2 n)$.

COMPLEJIDAD

Cada algoritmo de ordenamiento por definición tiene operaciones y cálculos mínimos y máximos que realiza (complejidad), a continuación una tabla que indica la cantidad de cálculos que corresponden a cada método de ordenamiento:

<u>Algoritmo</u>	<u>Operaciones máximas</u>
Burbuja	$\Omega(n^2)$
Inserción	$\Omega(n^2/4)$
Selección	$\Omega(n^2)$
Shell	$\Omega(n \log^2 n)$
Merge	$\Omega(n \log n)$
Quick	$\Omega(n^2)$ en peor de los casos y $\Omega(n \log n)$ en el promedio de los casos.

Referencias:

Tomado de la Universidad Nacional de La Matanza – UNLAM
Buenos Aires, Argentina

https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fwww.unlam.edu.ar%2Fdescargas%2F1_Metodos.doc%23%3A~%3Atext%3DM%25C3%25A9todos%2520de%2520Ordenamiento%26text%3DEs%2520la%2520operaci%25C3%25B3n%2520de%2520arreglar%2CAlg%25C3%25BAn%2520campo%2520en%2520un%2520registro.&wdOrigin=BROWSELINK

Fecha de Consulta: Abril de 2022