



Nombre:Alexia Castillo Díaz

Matricula:A00227250

Fecha: Julio 11,2022

Investigación

Importancia y eficiencia del uso de diferentes algoritmos de ordenamiento

Existen diferentes actividades en la vida diaria que requieren que los elementos se pongan en un orden específico, como por ejemplo las listas de estudiantes de una escuela que se pueden ordenar por apellidos o por los número de matrícula, estas tareas de ordenación son frecuentemente llevadas a cabo por las computadoras.

Los algoritmos de ordenamiento permiten ordenar la información de manera especial basandose en ciertos criterios, esto es muy importante al formar parte de criterios más complejos. Los elementos numéricos pueden ordenarse en orden creciente o decreciente de acuerdo al valor numérico, el elemento por el cual se ordenan datos se denomina clave.

Una colección de datos puede ser almacenada en un archivo, un vector o tabla, una lista enlazada o un árbol, si los datos estan en un vector o tabla es una ordenación interna si los datos se ordenan en un archivo es ordenación externa.

Además algunos algoritmos son más eficientes en cuanto al tiempo de ejecución considerando el factor constante o el orden de magnitud.

Para decidir que algoritmo utilizar se consideran 2 criterios: el que requiere menor tiempo de ejecución en computadora y el que necesita menor número de instrucciones, sin embargo no es fácil decidir ya que también interviene el estilo del programador

Existen diferentes algoritmos de ordenamiento entre los directos se encuentran los de burbuja, selección e inserción muy eficientes en listas sencillas y los indirectos que son shell, ordenación rápida, ordenación por mezcla radixsort necesarios para listas grandes

Ordenamiento por selección y reemplazo.

Es el más usado para arreglos pequeños y se encarga de ordenar los elementos de un arreglo nativo sin utilizar un segundo arreglo. El algrotimo se apoya en pasar

sucesivamente intercambiando el elemento más pequeño sucesivamente con el primer elemento de la lista en la primera pasada ordenándose el frente de la lista y el resto estará desordenado y en posteriores pasadas se irán ordenando los siguientes elementos más pequeños hasta que quede ordenada.

Algoritmo de la burbuja.

Consiste en realizar recorridos secuenciales en el arreglo intercambiando los elementos adyacentes que estén desordenados. Sin embargo al ordenar arreglos completamente desordenados el algoritmo de la burbuja toma el doble de tiempo que selección y reemplazo, pero puede llegar a ser más eficiente con arreglos semi-ordenados, sin embargo no es muy utilizado. En este tipo los valores pequeños suben en los vectores o tablas y los valores mayores se van a la parte inferior, se pasa varias veces por la tabla o vector y se comparan parejas de elementos, si una pareja está en orden creciente se deja como está si el orden es decreciente se intercambian sus valores.

Ordenamiento por inserción.

Es como ordenar tarjetas de nombres se va insertando el elemento en su posición correcta dentro de una lista o archivo ya ordenado.

Ordenación Shell.

Es una mejora a la inserción directa, donde cada elemento se compara con los elementos contiguos de su izquierda una tras otra, puede ser que se necesiten muchas comparaciones antes de que sea colocado en su lugar definitivo.

Quicksort.

Se trata de un algoritmo de ordenamiento recursivo y es el más eficiente conocido, se dice que quicksort con colas es mejor que la selección con arreglos nativos, sin embargo cuando se ordenan conjuntos de pocos elementos es más lento que con arreglos nativos, pero cuando los arreglos son de gran tamaño quicksort con colas es más rápido aunque no lo suficiente como lo es con arreglos nativos, pero es el algoritmo de ordenamiento más eficiente conocido.

Se basa en particiones de la lista a ordenar es posiblemente el más pequeño en código y más rápido, las listas se dividen eligiendo un elemento y se utiliza como elemento de partición, una vez elegido se utiliza para ordenar el resto de la lista en dos sublistas donde las claves son menores que el pivote o mayores e iguales, la lista final se obtiene concatenando la primera sublista, el pivote y la segunda sublista.

Reflexión personal.

Como he podido ver y poner en práctica los algoritmos de ordenamiento son una herramienta muy útil cuando hablamos de realizar tareas en las cuales sea

necesario ordenar los datos en orden ascendente, descendente, por orden alfabético, etc. Situaciones que son muy comunes en la vida diaria, como por ejemplo las listas de alumnos de una escuela, ordenar paquetes dentro de una empresa de mensajería por su código postal. A pesar de que los algoritmos son una herramienta muy útil no todos tienen la misma rapidez ni la misma complejidad en cuanto a instrucciones por lo que el uso de ellos dependerá de la situación particular en la cual se deseen utilizar y de las preferencias del programador el que pueda elegir una por encima de otra opción.

En el momento de utilizar los algoritmos de ordenamiento me he podido dar cuenta que para ordenar la información se realizan múltiples "pasadas" para ir ordenando los datos, es decir el proceso va acomodando la información poco a poco como sucede con el ordenamiento por selección y reemplazo, pero este no es único algoritmo de ordenamiento está el de burbuja, el de inserción, shell y el quicksort el cual es uno de los que se considera como más eficiente para arreglos de gran tamaño.

En el código de esta actividad utilizamos el método de la búsqueda binaria para poder encontrar las fechas que estaban entre las fechas que el usuario escribió. Este método lo que hace es dividir repetidamente a la lista para que los elementos de estas se conviertan en números y dependiendo de las fechas y los guarda en arreglos