



Tecnológico de Monterrey

Programación de estructuras de datos y algoritmos fundamentales
Profesor Luis Ricardo Peña Llamas
TC1031 Grupo 13

Act 1.3 - Actividad Integral de Conceptos Básicos y Algoritmos Fundamentales (Evidencia Competencia)

Carlos García Mercado A01633757

13 de septiembre del 2021

Investigación y explicación del programa

Introducción

Se tiene una bitácora de fallos para iniciar sesión en un sistema, en esta se indica el mes, día y hora en el que se intentó acceder, la dirección IP de la computadora de la que se intentó acceder, y el motivo por el que falló.

Esta bitácora está desordenada, y se tienen que usar métodos computacionales para ordenarla, así como permitir filtrar la información y acceder a ella solamente en un rango de tiempo.

Proceso del algoritmo

Se crea una clase Nodo para crear una lista doblemente ligada, es decir, se crea un vector donde cada elemento tiene un apuntador al siguiente elemento y al anterior.

Se le da un formato compatible a cada elemento compatible, se convierte el mes en un número de mes, y se crea un número con la fecha, el cual va a ser el que se usará para comparar.

Entero que se genera para comparar



Ahora se utiliza el algoritmo *bubble sort* para ordenar la fecha, llamado aquí `ordenarBurbuja()` y después se hace una búsqueda secuencial para encontrar la primera y última fecha que selecciona el usuario, para así mostrar solamente los elementos en el intervalo introducido.

Bubble Sort

Bubble Sort es un algoritmo de ordenamiento que pasa por cada elemento de la lista en el orden del índice, y compara dos elementos adyacentes, y los intercambia si están en el orden incorrecto, este proceso se repite hasta que la lista está completamente ordenada.

La eficiencia promedio de este algoritmo es de $O(n^2)$ por lo que no se considera un algoritmo eficiente. Sin embargo es fácil de entender y programar.

Búsqueda secuencial

La búsqueda secuencial, como Bubble Sort, recorre todos los elementos de la lista. En este se compara con cada elemento de la lista hasta encontrar el elemento igual. Si no se encuentra, se recorre toda la lista. En el peor caso, este es el que no se encuentra, su eficiencia es $O(n)$, y en el mejor es $O(1)$.

Complejidad computacional

Estas son la cantidad de declaraciones que tenemos:

De 359 líneas de código

23 declaraciones de if

03 declaraciones de for

04 declaraciones for dentro de otros for

Como se ignoran los constantes para determinar la complejidad computacional, y se hace caso al peor de los casos, la complejidad de este algoritmo es:

$$O(n^2)$$

En el peor de los casos, el tiempo de ejecución del algoritmo es el número de elementos al cuadrado.

Reflexión individual

Otros algoritmos

Dentro de los algoritmos de acomodo, Bubble Sort es uno de los algoritmos menos eficientes para el ordenamiento de arreglos, su eficiencia como ya se mencionó es de $O(n^2)$, es decir que el algoritmo va a ser mayor entre más elementos tenga el arreglo, por la cantidad de el elemento al cuadrado.

Sin embargo, este algoritmo es bueno porque es fácil de programar y debuggear, y en una base de datos relativamente pequeña como lo es esta, no resulta tan impráctico su uso.

Para ordenar nuestra base de datos, creamos un índice para cada fecha, que está basado el estándar ISO para fechas, pero guardado como un entero, de esta manera se puede hacer el programa que compare los enteros y los ordene de manera apropiada. Este sistema de indexación podría funcionar incluso con años, pero se tendría que añadir al inicio del entero, y se debe tomar en cuenta que es conveniente utilizar 4 dígitos para el año, para hacer así el programa útil para organizar fechas antes del 2000 y después del 2100

En general, este tipo de algoritmos son importantes en nuestro día a día, ya que encuentran la manera más fácil de ordenar y buscar entre diferentes listas de cosas, y es gracias a

estos que actualmente tenemos servicios como Google, y no sólo que los tenemos, pero que son tan rápidos como lo son, y no sólo ahí, pero dentro de nuestras computadoras y celulares como son creados diferentes índices para los archivos, y así pueden ser ordenados de manera más rápida y luego buscar entre ellos, y son la razón por la que usando, por ejemplo, spotlight search en Mac se puede buscar el nombre de un archivo o la fecha, y los puede encontrar de manera rápida.