

Trabajo Práctico Especial

Etapa 1

Programación 3

TUDAI

Facultad de Ciencias Exactas

Universidad Nacional del Centro de la Provincia de
Buenos Aires

Integrantes:

Miguez, Jose Maria (josemiguez148@gmail.com)
Tubino, Bruno Gonzalo (bruno.tubino33@gmail.com)

Entrega:

4 de Mayo del 2018

Introduccion

La primer entrega del TPE consiste en implementar la lógica necesaria para obtener una colección de libros que contenga un género en particular, ingresado por el usuario.

La herramienta comenzará llevando a memoria la colección completa de libros para luego realizar un filtrado por un género dado, presentando al usuario la colección de libros resultante.

Para optimizar el proceso de búsqueda, se requiere implementar un índice por género, el cual simplificará el acceso a solo un subconjunto de todos los libros existentes.

Elección y justificación de estructura de almacenamiento

Estuvimos teniendo en cuenta la posibilidad de utilizar ArrayList , pero esta tiene complejidad ($O(n)$) cada vez que inserto un elemento al principio se tiene que hacer un corrimiento de todos los elementos y esto no es lo más eficiente.

Decidimos elegir una LinkedList para almacenar los libros en memoria , debido a que tiene una complejidad constante ($O(1)$) porque los elementos se insertan o al principio o al final y no es necesario ningún corrimiento.

Discusión de estructuras de índice de acceso por género

Lista simplemente vinculada

La lista para realizar una búsqueda binaria, siempre se tiene que volver al ingresar al primer nodo que es el que tiene el puntero. Esto tiene un gran costo, ya que se tiene que recorrer la lista muchas más veces.

ArrayList

En un ArrayList tenemos que primero encontrar el medio del mismo , para eso recorrerlo. Y una vez que estamos posicionados podemos empezar a comparar.

Un árbol binario de búsqueda

En un ABB al acceso ya se posiciona por el medio del mismo y a partir de ahí comienza la búsqueda binaria, la cual tiene complejidad de $(\log(2)n)$.

Elección de estructuras de índice de acceso por género

Consideramos que la búsqueda en este tipo de árboles es muy eficiente , representa una función logarítmica. El máximo número de comparaciones que necesitamos para saber si un elemento se encuentra en un árbol binario de búsqueda estaría entre $[\log_2(N+1)]$ y N , siendo N el número de nodos.

El grafico representa la comparación entre LinkedList y ArrayList al momento de insertar un elemento.

Conclusión

En este práctico pudimos visibilizar las diferentes ventajas que tienen las estructuras a la hora de realizar ciertas opciones, por ejemplo a la hora de buscar el más eficiente es el ABB.

La mayor dificultad que tuvimos , fue poder separar los datos en diferentes estructuras para después manipularlos de forma eficiente (y hacer el grafico)