30/06/2022

**TRABAJO PRÁCTICO ESPECIAL**

**PROGRAMACIÓN 3**

Entrega N°2

TUDAI- EXACTAS - UNICEN

Autores:

* De Francesco German (*germandefrancesco@gmail.com*)
* Ballone Gabriel (*gabrielgeno@gmail.com*)

# Introducción

Este trabajo prácrito se planteó en dos etapas, para las cuales se presentaron una serie de distintas consignas.

Para la **primera etapa**, dado un conjunto de diferentes *datasets,* se requiere 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.

Entrada: El programa tomará un archivo .*csv* de entrada, con un formato como el que se muestra a continuación.

*{Titulo,Autor,Paginas,Generos}*

Salida: El programa deberá generar un archivo *.csv* con los títulos de los libros que cumplen con el género dado.

Consignas

1. Discutir y analizar el costo de las operaciones sobre las distintas estructuras que se podrían utilizar para almacenar los libros en memoria.

* Justificar la elección de la estructura considerando la utilización de la misma dentro de la herramienta.

1. Considerando las siguientes estructuras como posibles índices de acceso por género:

* Una lista simplemente vinculada (implementada en el práctico 1).
* Alguna de las implementaciones conocidas de la interfaz List de Java.
* Un árbol binario de búsqueda.

1. Discutir y analizar el costo para realizar un filtrado utilizando cada una de las estructuras propuestas como índice de acceso.
2. Implementar el índice de acceso por géneros eligiendo una de las estructuras. Justificar su elección.
3. Realizar pruebas de ejecución utilizando los distintos .cvs provistos y analizar los resultados, considerando distintas métricas (tiempo de ejecución, cantidad de iteraciones, cantidad de nodos visitados, etc.).

En la **segunda etapa** se hace foco en las estadísticas de búsqueda.

Se desea realizar un análisis de la utilización del buscador, por parte de los usuarios; puntualmente la relación entre los géneros ingresados en las distintas búsquedas. Se asume que la herramienta permite ingresar un conjunto de categorías a buscar, con lo cual la colección de libros resultantes contendrá sólo los libros que cumplen con todas las categorías ingresadas.

Se proveerá como entrada al programa varios archivos .csv con los sucesivos géneros que se ingresaron en distintas búsquedas realizadas. A partir de esta información se desea construir un grafo, donde:

* Cada vértice representa un género que fue incluido en alguna búsqueda; y
* Un arco que comunica dos vértices indicará la cantidad de veces que luego de buscar el primer género inmediatamente a continuación se buscó por el segundo género.

Utilizando este grafo como estructura se deberán implementar los siguientes servicios:

* Obtener los N géneros más buscados luego de buscar por el género A.
* A partir de un género A encontrar, en tiempo polinomial, la secuencia de géneros que más alto valor de búsqueda posee. Vamos a definir el valor de búsqueda de una secuencia como la suma de los arcos que la componen.
* Obtener el grafo únicamente con los géneros afines a un género A; es decir que, partiendo del género A, se consiguió una vinculación cerrada entre uno o más géneros que permitió volver al género de inicio.

Entrada:

ensayo,ficción,fantasía,cine

fotografía,periodismo,humor,relatos,tecnología

poesía,tecnología

fotografía,ficción

fotografía,ficción,investigación

humor,relatos,arte

Consigna

- Implementar el grafo que permita registrar las relaciones entre géneros.

* Detallar las estructuras elegidas para implementar el grafo.

- Implementar los servicios indicados.

- Realizar un análisis teórico y empírico de los servicios implementados.

* Para ambos análisis se debe elegir alguna/s métrica/s que permita entender el costo

de cada algoritmo para resolver el servicio planteado.

* Analizar si las estructuras elegidas para implementar el grafo tienen un impacto

positivo o negativo en el costo de los servicios.

# Desarrollo del trabajo

* ***Etapa 1***

Se optó por utilizar una lista vinculada simple (LinkedList) tanto para el conjunto de libros como así también para el conjunto índice de géneros.

En el caso de los libros, dado que la consigna sólo solicita obtener todos los libros de un género, es decir, no hay que buscar por título o algo que requiera algún criterio de ordenamiento, la búsqueda siempre será lineal, de principio a fin (O(n)). Utilizando esta estructura estamos más que cubiertos y no tenemos que traer todos los libros a memoria (realizando un malgasto de ésta) como sería utilizando un ArrayList.

Además, la búsqueda no requiere recorrer la lista en forma reversa o acceder a un libro en particular, lo cual sería una de las desventajas de LinkedList (a menos que sea doblemente vinculada, pero esto representa mayor uso de memoria para guardar el puntero extra).

Una de las ventajas de las LinkedLists es que, al ser una estructura dinámica, puede crecer en tiempo de ejecución en base a la asignación y des asignación de memoria, por lo que no hay necesidad de darle un valor inicial.

También cabe mencionar que su implementación es más sencilla a pesar de que, en nuestro caso, estamos utilizando una estructura provista por las librerías de Java.

En el caso de los géneros, implementamos este mismo tipo de estructura dado que son un conjunto de tamaño reducido y la diferencia entre un tipo de estructura y otro no implica una diferencia notable.

Una solución más rápida habría sido utilizar un árbol binario de búsqueda. Sin embargo, según la consigna nos interesa un género en particular y encontrarlo en un ABB implicaría, en el peor de los casos, recorrer la mitad de sus elementos (O(h), siendo h la altura del árbol). Pero, como se mencionó anteriormente, al ser un conjunto acotado no va a presentar una diferencia notable y la implementación es más compleja.

* ***Etapa 2***

En esta etapa comenzamos definiendo la interface de base “*Graph”* y la clase que la implementa *DirectedGraph* junto con la clase *Arc* que permiten plasmar el grafo dirigido que solita la consiga para registrar las relaciones entre géneros.

En la clase *DirectedGraph* utilizamos la estructura HashSet que provee Java para almacenar los vértices y arcos que componen el grafo ya que por definición esta estructura al implementar una instancia de Set no permite almacenar elementos repetidos y al operar sobre una tabla de hash, ofrece un costo de tiempo constante en operaciones básicas como “add”, “remove”, “contains” y “size” por ejemplo. Luego la clase Arc almacena los vértices de origen y destino y el valor del mismo

Para cada cumplir con cada uno de los servicios solicitados se implementaron los siguientes métodos:

* *“Obtener los N géneros más buscados luego de buscar por el género A.”*

**LinkedList<String> getMostSearchedGenresAfterAGenre**

**(DirectedGraph<Integer> graph, String genre, int amount)**

* *“A partir de un género A encontrar, en tiempo polinomial, la secuencia de géneros que más alto valor de búsqueda posee. Vamos a definir el valor de búsqueda de una secuencia como la suma de los arcos que la componen.”*

**LinkedList<String> getSequenceWithHigherSearchRatio**

**(DirectedGraph<Integer> graph, String genre)**

* *“Obtener el grafo únicamente con los géneros afines a un género A; es decir que, partiendo del género A, se consiguió una vinculación cerrada entre uno o más géneros que permitió volver al género de inicio.”*

**DirectedGraph<Integer> getGraphWithAffinedGenres**

**(DirectedGraph<Integer> graph, String genre)**

El análisis que podemos hacer de los servicios implementados realizamos los siguientes

# Conclusiones

Es importante saber o poder analizar el tipo de estructura con el cual se va a trabajar o procesar una solución. Es algo que a veces no se tiene en cuenta pero que a la larga o al crecer el proyecto puede traer consecuencias no deseadas en el rendimiento.