

Estructura de Datos y Algoritmos: Laboratorio 3

Profesor: Yerko Ortiz
Ayudante Lab: Diego Banda



Información importante

Enunciado y rúbrica

Disponible en Canvas, sección
módulos.

Fecha de entrega

Viernes 14 de junio a las 23:59



Código a usar

```
1 class Cancion {  
2     public int id;  
3     public String nombre;  
4     public String artista;  
5     public double duracion; // en milisegundos  
6     public Cancion(String nombre, String artista, double duracion){  
7         this.nombre = nombre;  
8         this.artista = artista;  
9         this.duracion = duracion;  
10    }  
11}
```



Funciones

**agregarCancion
(cancion)**

Recibe una canción y la agrega a la base de datos. La base de datos es un arreglo o lista.

**ordenarPor
Nombre(songs)**

Toma una base de datos de canciones (arreglo o lista) y devuelve la misma estructura en orden lexicográfico.

**ordenarPor
Duracion(songs)**

Toma una base de datos de canciones (arreglo o lista) y devuelve la misma estructura en orden lexicográfico.

Lexicográfico: En matemáticas, o más particularmente en Teoría del Orden, el orden lexicográfico es una relación de orden definida sobre el producto cartesiano de conjuntos ordenados. Es conocido principalmente por su aplicación a cadenas de caracteres, por ejemplo en diccionarios o en la guía telefónica.

Ordenamiento

Sorting algorithms	Time complexity		
	Best	Average	Worst
Bubble sort	$O(n)$	$O(n^2)$	$O(n^2)$
Insertion sort	$O(n)$	$O(n^2)$	$O(n^2)$
Merge sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$
Quick sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$

Los métodos de ordenamiento deben ser implementados usando 4 formas: MergeSort, QuickSort, InsertionSort y BubbleSort. Tanto para el orden por nombre como por duración, genere una tabla midiendo el tiempo de ejecución de cada método y comente los resultados.

Funciones

crearArbol (songs)

Recibe un arreglo o lista de canciones y crea un árbol binario. Cree un árbol a partir del arreglo usando búsqueda binaria (lo que resulta en un árbol balanceado) y otro árbol desde un arreglo sin ordenar.

buscarCancion (String nombre)

Recibe el nombre de una canción y devuelve el ID de esta, si no existe, retorna -1. Busque en los árboles creados anteriormente.

buscarCancion (double duracion)

Recibe una duración en milisegundos y devuelve el ID de las canciones con esa duración como un arreglo o lista. En caso de no haber ninguna canción con esa duración retorna el arreglo o lista vacío. Busque en los dos árboles creados anteriormente.

Tiempo

```
1 long start, finish;  
2  
3 start = System.nanoTime();  
4 dataMergeSort = SortingMethods.MergeSort(dataMergeSort); //Ordena un arreglo  
5 finish = System.nanoTime();  
6 System.out.println("MergeSort time: " + (finish - start) + " ns.");
```

Ejecute un número Q de queries sobre el árbol ordenado y el desordenado. Comente los resultados.



Archivo CSV



<https://github.com/DiegoBan/EDDA2024-1-S4/tree/main/Laboratorios/Lab3/CSV>