

Mergesort y Comparación de Algoritmos de Ordenamiento

1. Descripción de la Actividad

El objetivo del laboratorio es realizar la implementación del algoritmo de ordenamiento Mergesort, hacer una comparación de éste con los algoritmos de ordenamiento Insertionsort, Bubblesort y Selectionsort y resolver un ejercicio donde aplique los conocimientos adquiridos.

En el archivo ordenamiento.py del laboratorio 1 deberá agregar la implementación del algoritmo de ordenamiento Mergesort. Los archivos que deberán incluir al momento de entregar el laboratorio son:

- ordenamiento.py: Librería que debe contener la implementación de los algoritmos de ordenamiento.
- arrayT.py: Estructura de datos arreglo a utilizar.
- ord.py: Algoritmo de ordenamiento utilizado para resolver el problema del ejercicio 3.
- libros.py: Cliente para el ejercicio 3.
- Gráficas generadas.

2. Detalles de la Actividad

- Ejercicio 1: Se debe implementar el algoritmo de ordenamiento Mergesort siguiendo el pseudocódigo presentado en el libro de Kaldewaij.
- Ejercicio 2: Se desea estudiar el tiempo de ejecución de los algoritmos. Debe ejecutar con al menos 5 arreglos de diferentes tamaños, tal que estos sean representativos, es decir, tamaños con los cuales se pueda observar que el tiempo obtenido por los algoritmos se corresponde con lo esperado en la teoría. Para cada prueba, es decir, para cada número de elementos del arreglo, debe realizar al menos tres corridas. Deberá realizar las gráficas de dichas pruebas y explicar los resultados obtenidos. Dicha explicación la puede colocar como comentario al final del archivo ordenamiento.py.
- Ejercicio 3: Los estudiantes del Laboratorio de Algoritmos II deben estudiar muy arduamente para salir bien en sus parciales. Es por ello que tienen una lista con los nombres de los libros y sus autores para todos los días poder hacer al menos 100 ejercicios y salir bien. Para poder llevar un control de los libros que ya usaron, los estudiantes deciden ordenar la lista de dos maneras: una en orden alfabético por autor y la otra en orden alfabético por título. Como los estudiantes quieren que su programa se ejecute lo más rápido posible, al momento de implementarlo usarán de los cuatro algoritmos de ordenamiento que han aprendido aquel que sea de mayor eficiencia. El

algoritmo de ordenamiento se encontrará en un archivo llamado `ord.py` y recibirá como parámetros dos arreglos. Adicionalmente, se deberá crear un archivo cliente llamado `libros.py` y éste le pedirá al usuario el archivo con los libros y autores que desea procesar. Dicho archivo de entrada tendrá el siguiente formato:

```
n
Autor_1[TAB]Libro_1
Autor_2[TAB]Libro_2
Autor_3[TAB]Libro_3
...
Autor_n[TAB]Libro_n
```

donde en la primera línea n es el número de pares autor y libro, para luego en las siguientes n líneas se presentan los autores y los libros, uno por línea separados por tabulador. La salida deberá colocarse en un archivo y deberá tener el siguiente formato:

```
LibroOrd_1[TAB]Autor_1
LibroOrd_2[TAB]Autor_2
LibroOrd_3[TAB]Autor_3
...
LibroOrd_n[TAB]Autor_n

AutorOrd_1[TAB]Libro_1
AutorOrd_2[TAB]Libro_2
AutorOrd_3[TAB]Libro_3
...
AutorOrd_n[TAB]Libro_n
```

donde `Ord` indica que los elementos están ordenados por libro o autor.

Características a tomar en cuenta en la implementación:

- Las columnas en ambos archivos se encuentran separadas por un tabulador.
- Podrá haber tantos libros como se quisiese.
- Solo se encontrará el apellido del autor y el nombre del libro en el archivo de entrada. El nombre del autor no aparecerá en el archivo.
- En caso de existir un mismo autor con libros diferentes se deberá ordenar también por el orden lexicográfico de los libros.
- La primera letra del libro o del autor podría estar en minúscula.

Un ejemplo de entrada y salida sería el siguiente:

Entrada:

```
6
Castellanos Apuntes de Discretas III
Gries A Logical Approach to Discrete Math
```

Bonet Láminas de Algoritmos y Estructuras de Datos II
Cormen Introduction to Algorithms
Mart Learning Python
Guido Style Guide for Python Code

Salida:

Bonet Láminas de Algoritmos y Estructuras de Datos II
Cormen Introduction to Algorithms
Castellanos Apuntes de Discretas III
Mart Learning Python
Gries A Logical Approach to Discrete Math
Guido Style Guide for Python Code

A Logical Approach to Discrete Math Gries
Apuntes de Discretas III Castellanos
Introduction to Algorithms Cormen
Láminas de Algoritmos y Estructuras de Datos II Bonet
Learning Python Mart
Style Guide for Python Code Guido

- **Observación:** debe usar como arreglo en todas las implementaciones, la estructura de datos definida en el módulo `arrayT.py`, tal como se hizo en el laboratorio de la semana 2.

3. Condiciones de Entrega

El trabajo es por equipos de laboratorio. Debe entregar los códigos fuentes de sus programas y las gráficas, en un archivo comprimido llamado `LabSemana3-X-Y.tar.gz` donde X y Y son los números de carné de los integrantes del grupo. La entrega se realizará por medio del aula virtual antes de la 4:30 pm del jueves 26 de Enero de 2017.