

Práctico 13

Ordenación

1. Ordene los siguientes números 76, 21, 34, 68, 31, 27, 53 utilizando el algoritmo de ordenación por Inserción (*Insertion Sort*)
2. Ordene los siguientes números 6, 2, 4, 8, 3, 7, 5 utilizando el algoritmo de ordenación por Selección. (*Selection Sort*).
3. Pida al usuario 6 números y muestrelos ordenados, utilizando los métodos por Selección, Inserción y Burbuja.
4. Dado un archivo de texto llamado *nros.txt* (*Puede generarlo desde el mismo código si así lo prefiere*) que contendrá un número por línea, 50 en total, generar un nuevo archivo, llamado *ordenado.txt* que tendrá los números ordenados. Utilizar el algoritmo de ordenación que prefiera.
5. Dado el algoritmo de ordenación que usted prefiera, realice una segunda implementación del mismo, para parámetros del tipo array en vez de listas. Tomará un array de elementos a ordenar, devolverá un array ordenado. ¿Notó diferencias? ¿En qué?
6. Genere un arreglo con 10 números al azar entre 1 y 100, y ordenelos de menor a mayor. Utilice al menos tres métodos de ordenación diferentes.

Variadoo

7. Dada la siguiente lista [*'barco', 'casa', 'ala'*] ordenela con el método de la Burbuja o el de Selección. ¿Es posible ordenar cadenas de caracteres? ¿Porqué?
8. Implemente un programa que pida al usuario 10 precios de golosinas. Ordene los precios de menor a mayor utilizando al menos 2 métodos diferentes.
9. Genere una lista de 6 números aleatorios y un array de 1 dimensión, con 6 números aleatorios. Ordene ambos con al menos 2 algoritmos de ordenación diferentes.
10. Genere un array de 4x6 números enteros, y ordene los valores solamente en cada fila. Ejemplo, para la siguiente matriz A de 3x3, se retornaría la siguiente matriz ordenada O:

$$A_{3 \times 3} = \begin{pmatrix} 2 & 5 & 0 \\ 7 & 3 & 8 \\ 3 & 0 & 1 \end{pmatrix} \quad O_{3 \times 3} = \begin{pmatrix} 0 & 2 & 5 \\ 3 & 7 & 8 \\ 0 & 1 & 3 \end{pmatrix} \quad (1)$$

11. Dado un archivo de texto, llamado *pacientes.txt* que contendrá los datos de un paciente, por fila, a saber: apellido, nombre, edad, obra social. Deberá leer los datos del archivo y mostrarlos ordenados por edad. Además, deberá guardar todos los datos en un nuevo archivo llamado *pacientes_por_edad.txt* A continuación se muestra un ejemplo de cómo estarían los datos en el archivo *pacientes.txt*

Sanchez, Luis, 64, OSDE
Balaz, Sara, 32, OSECAD
Lipa, Julieta, 27, OSDE
Perez, Lucila, 30, OSECAD

12. Se tiene un libro en formato archivo de texto. Obtenga del mismo, 50 palabras en total, del siguiente modo: Una palabra al azar de cada línea, hasta completar las 50 palabras en total. Luego, ordenelas utilizando al menos dos algoritmos de ordenación diferentes.
13. Genere del modo que prefiera, un archivo de texto llamado *palabras.txt* que contendrá una palabra por línea. Con un total de al menos 10 palabras. Muestre la última palabra, en orden alfabético.
14. Pida al usuario ingresar 10 notas de examen de alumnos. Muestre la peor y mejor nota.
* Sin utilizar *max* y *min* de Python
15. Dada una lista de 10 nombres cualesquiera, ordenelos respecto a sus longitudes, en vez de alfabéticamente. Por ejemplo, los siguientes nombres quedarían:

Desordenados: "Legolas", "Sam", "Frodo", "Sauron", "Gollum"

Ordenados: "Sam", "Frodo", "Sauron", "Gollum", "Legolas"

16. Mida para los distintos algoritmos de ordenación aprendidos, utilizando alguna lista de números, sus tiempos de ejecución en cada caso y compare. Realice las comparaciones, utilizando primero una lista de números pequeña, por ejemplo de 10 números. Luego utilizando una lista de números grande, por ejemplo, de 6 millones de números. ¿Cuáles algoritmos tardaron más tiempo, y cuáles fueron más rápidos?