

UNIVERSIDAD DE GUADALAJARA

Centro Universitario de Ciencias Exactas e Ingenierías

División de Tecnologías para la Integración Ciber-Humana



Análisis de Algoritmos

Mtro. Jorge Ernesto Lopez Arce Delgado

Elizabeth Arroyo Moreno, Código: 221453749

Karla Rebeca Hernández Elizarrarás, Código: 223991977

Ingeniería en Computación

Act. Visualizador de métodos de Ordenamiento

Septiembre de 2025

Introducción

Para esta práctica agregamos más métodos de ordenamiento a nuestra interfaz, desarrollando más herramientas al programa, tales como la selección de un método distinto de ordenamiento, la selección personalizada de número de datos junto con la velocidad en la que se realiza la búsqueda. usando una nueva librería para nuestra GUI customtkinter. Para poder comparar la lógica de los 4 algoritmos que agregamos.

Objetivo

- Seleccionar el algoritmo desde un dropdown.
- Generar datos, Ordenar, Mezclar (shuffle), y Limpiar resaltado con botones.
- Modificar el número de barras desde un Entry.
- Ajustar la velocidad de la animación con un Scale.
- Mejorar la GUI

Desarrollo

Utilizando el código pasado comenzamos a agregar primero los nuevos algoritmos de búsqueda, como el código no es modular, pero si tiene un orden en cuanto a la estructura del código, los nuevos algoritmos se encuentran bajo el primero de selección, los algoritmos que se añadieron fueron tomados del github que el profesor nos compartió del cual solo cambiamos las especificaciones y el uso del yield en la participación pasada.

```
40 # ALGORITMO: BUBBLE SORT
41 def bubble_sort_steps(data, draw_callback):
42     n = len(data)
43     for i in range(n - 1):
44         for j in range(0, n - i - 1):
45             draw_callback(activos=[j, j + 1])
46             yield
47             if data[j] > data[j + 1]:
48                 data[j], data[j + 1] = data[j + 1], data[j]
49                 draw_callback(activos=[j, j + 1])
50             yield
51     draw_callback(activos=[])
52
53 # ALGORITMO: MERGE SORT
54 def merge_sort_steps(data, draw_callback, inicio=0, fin=None):
55     if fin is None:
56         fin = len(data)
57
58     if fin - inicio > 1:
59         medio = (inicio + fin) // 2
60         yield from merge_sort_steps(data, draw_callback, inicio, medio)
61         yield from merge_sort_steps(data, draw_callback, medio, fin)
62
63         izquierda = data[inicio:medio]
64         derecha = data[medio:fin]
65         i = j = 0
66         k = inicio
67
68         while i < len(izquierda) and j < len(derecha):
69             draw_callback(activos=[k])
70             yield
71             if izquierda[i] <= derecha[j]:
```

Se colocaron funciones de la nueva librería que utilizamos para mejorar la visualización de la interfaz haciéndola más llamativa. Agregamos funciones junto con sus respectivos botones para la interfaz, los cuales el único que ya teníamos del código pasado era el de generar datos y ordenarlos con su respectivo algoritmo, añadimos un menú como en nuestra primera actividad pero esta vez con los algoritmos disponibles en el código, adicionamos botones también funciones como mezclar, un entry que te permite elegir el número de barras a ordenar, además implementamos un scale de 0-200

ms para la ajustar la velocidad en la que se realiza el ordenamiento.



N Datos	Velocidad	Selección	Burbuja	Merge	Quick
15	25	47s	65s	10s	9s
90	5	57s	74s	10s	7s

Conclusión

Esta actividad nos facilitó la visualización de los algoritmos de búsqueda, el funcionamiento de ellos y comprender más su lógica así como complejidad de cada uno, al modificar la velocidad en la animación de la gráfica es más fácil observar este cambio y comparar los métodos de búsqueda, también la cantidad de elementos en la gráfica nos hace notar esta diferencia. En agrupaciones más pequeñas los algoritmos como el de selección o el de burbuja funcionan con mayor eficacia y menor complejidad pero eso ya no sucede con listas mayores.

Bibliografías

Bratley, P., & Brassard, G. (1997). Fundamentos de algoritmia.

Levitin, A. (2008). Introduction to design and analysis of algorithms, 2/E. Pearson Education India.

Python Cheatsheet. (s.f.). Python GUI reference. Python Cheatsheet. <https://scouv.lisn.upsaclay.fr/python-memento/index.fr.html>