

UNIVERSIDAD DE GUADALAJARA



CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

Seminario de Algoritmia

Reporte de práctica

Nombre del alumno:	Ricardo David López Arellano
Profesor:	Erasmus Gabriel Martínez Soltero
Título de la práctica:	Tarea 3. Video de comparación
Fecha:	15 febrero del 2023

Capturas de pantalla del código

```
# -*- coding: utf-8 -*-  
"""  
Created on Tue Feb 14 23:50:47 2023  
  
@author: David López  
"""  
  
import random  
import numpy as np  
import matplotlib.pyplot as plt  
import time  
import matplotlib  
from matplotlib.animation import FFMpegWriter  
import matplotlib.animation as manimation  
import os  
  
matplotlib.use('TkAgg')  
random.seed(10)  
np.random.seed(10)
```

Librerías

```
def Burbuja(lista, elemento):  
    movimiento = 0  
    for n in range(len(lista) - 1, 0, -1):  
        for i in range(n):  
            if lista[i] > lista[i + 1]:  
                movimiento += 1  
                temp = lista[i]  
                lista[i] = lista[i + 1]  
                lista[i + 1] = temp  
    return movimiento
```

Función para la búsqueda de burbuja

```
def merge_sort(lista,elemento):
    k = 0
    if len(lista) > 1:
        mitad = len(lista) // 2
        primera_mitad = lista[:mitad]
        segunda_mitad = lista[mitad:]

        merge_sort(primer_mitad)
        merge_sort(segunda_mitad)
        i = 0
        j = 0

        while i < len(primer_mitad) and j < len(segunda_mitad):
            if primera_mitad[i] < segunda_mitad[j]:
                lista[k] = primera_mitad[i]
                i += 1
            else:
                lista[k] = segunda_mitad[j]
                j += 1
            k += 1

        while i < len(primer_mitad):
            lista[k] = primera_mitad[i]
            i += 1
            k += 1

        while j < len(segunda_mitad):
            lista[k] = segunda_mitad[j]
            j += 1
            k += 1

    return k
```

Función para la búsqueda de MergeSort

```
FFMpegWriter = manimation.writers['ffmpeg']
metadata = dict(title='Burbuja vs MergeSort', artist='David López',comment='Tarea 3 :')
writer = FFMpegWriter(fps=24, metadata=metadata)
#fig = plt.figure()
fig, (ax1, ax2) = plt.subplots(1, 2,figsize=(20,10))
ax1.title.set_text('Burbuja')
ax2.title.set_text('MergeShort')
ax1.set_ylabel("Consultas")
ax1.set_xlabel("# Elementos")
ax2.set_xlabel("# Elementos")
with writer.saving(fig, "BurbujaVsMergeShort.mp4", 100):
```

Proceso de mi función para graficarlas y guardar el video

```
plt.ion()
for idx,elementNum in enumerate(elementNumArray):
    lista=np.linspace(0,100000,elementNum))

    start=time.time()
    it=Burbuja(lista)
    finish=time.time()
    iteracionesBi.append(it)
    tiemposBi.append(finish-start)

    start=time.time()
    it=merge_sort(lista)
    finish=time.time()
    iteracionesLi.append(it)
    tiemposLi.append(finish-start)

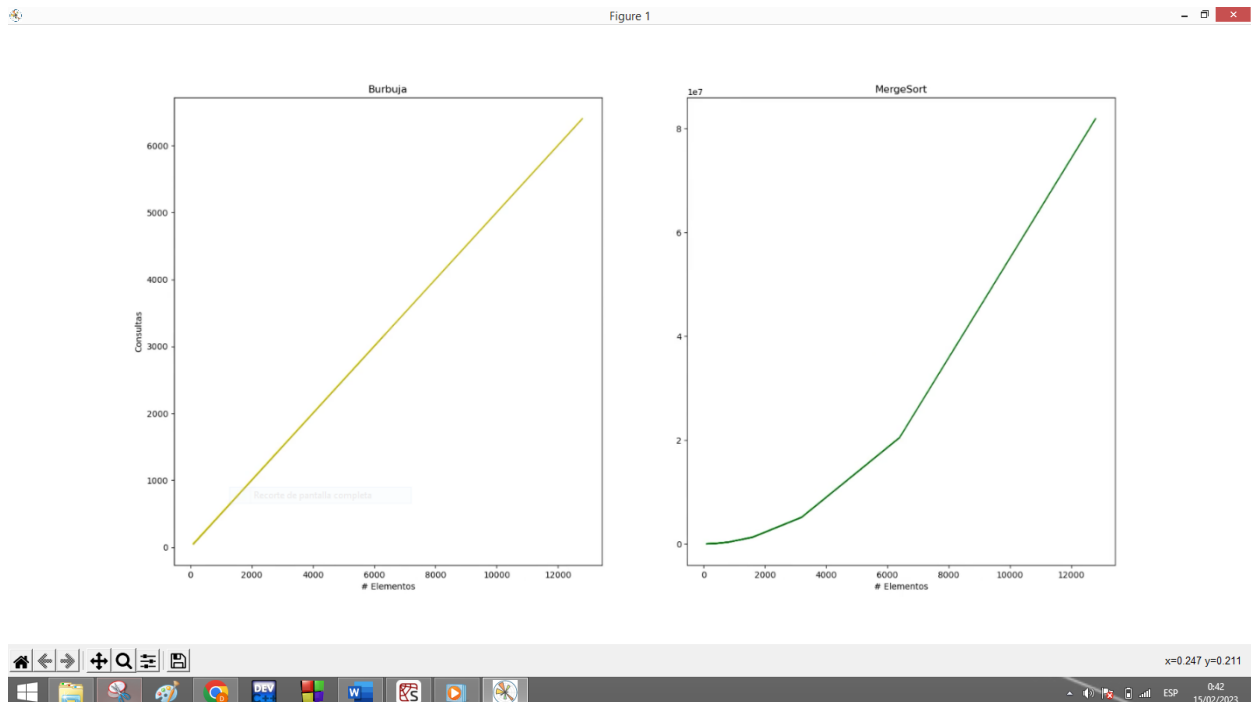
    ax1.plot(elementNumArray[:idx+1], iteracionesLi, 'r-',label='Burbuja')
    ax2.plot(elementNumArray[:idx+1], iteracionesBi, 'b-',label='Merg Sort')

    fig.canvas.draw()
    fig.canvas.flush_events()
    plt.show(block=False)

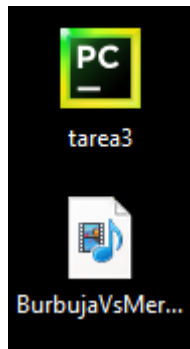
    time.sleep(.1)
    for i in range(12):
        writer.grab_frame()
```

Imprimir las gráficas y que vaya cambiando constantemente

Resultados



Estas son las gráficas en video que me fueron resultantes



De esta manera se me guardó el video con el nombre ingresado en el código

Conclusiones

En esta práctica usamos los métodos de burbuja y Merge Sort los cuales son muy efectivos para realizar búsquedas, el método burbuja consiste en comparar cada elemento de la estructura con el siguiente e intercambiándolos si corresponde y el método de Merge Sort consiste en partir una estructura en mitades, ordenar cada mitad y luego intercalar ordenadamente ambas mitades. Cada mitad se ordena aplicando el mismo método. También puedo decir que aprendí a graficar guardándolo en un video.