



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E ING

CUCEI

Análisis de Algoritmos

Act01 - Búsqueda con GUI

ALUMNO:

Angel Miguel Villalvazo Vazquez | 220653671

MAESTRO:

Jorge Ernesto López Arce Delgado

19/08/2025

Introducción

Se desarrolló un programa para medir los tiempos de ejecución de dos algoritmos de búsqueda implementados en el archivo algoritmos.py: búsqueda lineal y búsqueda binaria. Para mostrar unos resultados más estables, se usaron 5 tamaños de entrada y 5 repeticiones por cada tamaño. Los resultados muestran los tiempos promedios sobre el tiempo medido en segundos.

Descripción

Funciones utilizadas (archivo algoritmos.py):

- **generar_datos(n)**: genera una lista/array con una longitud n números enteros aleatorios.
- **busqueda_lineal(lista, x)**: búsqueda secuencial.
- **busqueda_binaria(lista, x)**: búsqueda binaria (más un ligero retraso debido a la función sort aplicada a "lista").



Tamaños probados (n): 100, 1,000, 10,000, 100,000, 1'000,000.

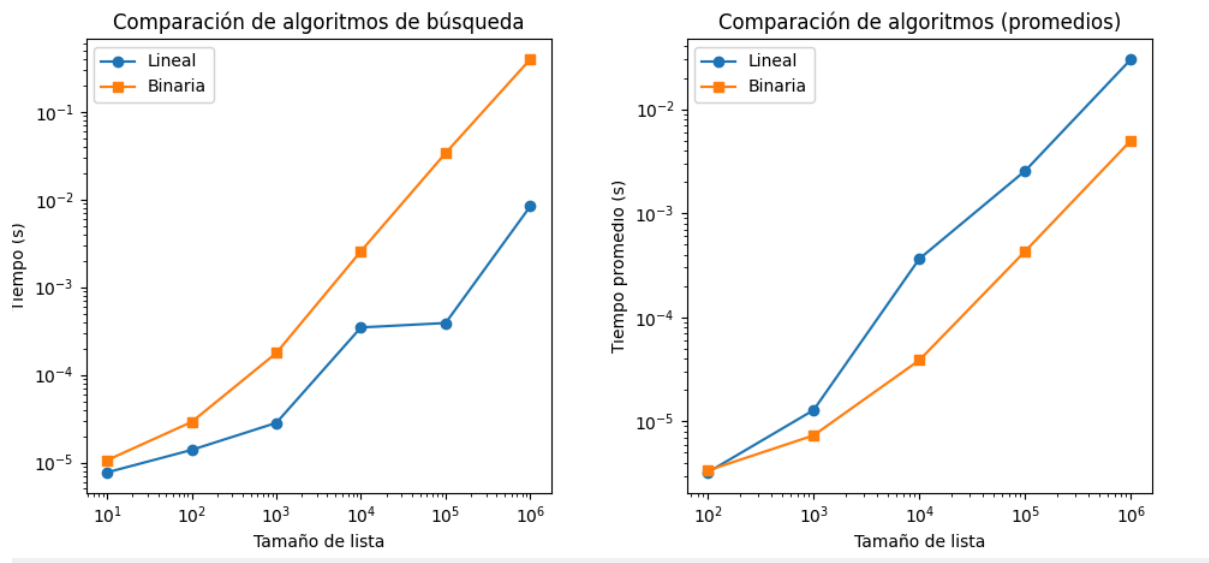
Repeticiones por tamaño: 5 (se promediaron los tiempos de las 5 repeticiones).

Selección del objetivo: en cada repetición se eligió X como un elemento aleatorio presente en la lista (np.random.choice(datos)), por lo que se mide el caso "elemento presente".

Medición: time.perf_counter().

Interpretación de los resultados

Si analizamos las gráficas podemos observar como la búsqueda lineal, efectivamente cuenta con un coste de tiempo lineal, sin embargo, con la búsqueda binaria no pasa lo que se esperaría (una función logarítmica), esto debido a que no solo estamos midiendo el tiempo en el que busca sobre la lista, sino también, el tiempo que tarda para ordenar dicha lista.



En dado caso de que la lista se ordenara una sola vez y después se midiera únicamente lo que tarda la búsqueda binaria, en teoría deberíamos esperar que cumpla con el comportamiento de $O(\log(n))$.

Conclusión

Con las gráficas actuales nos podemos dar cuenta que la combinación de búsqueda binaria más ordenación siempre termina siendo más cara en cuestiones de tiempo que la búsqueda lineal. Aunque si realizamos varias ejecuciones del programa con las diferentes cantidades de datos, podemos ver como la curva de la búsqueda binaria se encuentra casi en todo momento por debajo de la línea de búsqueda lineal.