

Reporte

Análisis a los algoritmos de búsqueda

Alumno: Lopez Esparza Angel Emanuel

Código: 223991772

Materia: Análisis de algoritmos

Sección: D06

Profesora: Jorge Ernesto Lopez Arce Delgado



Introducción

El reporte analiza el tiempo de ejecución de 2 algoritmos de búsqueda, el algoritmo de búsqueda lineal y el algoritmo de búsqueda binario.

Los algoritmos a analizar son los siguientes:

```
for reps in range(REPETICIONES):
    tiempo_inicial = time.perf_counter()
    for i, _dato in enumerate(interfaz_datos[0]):
        if dato == _dato:
            break
    tiempo_final = time.perf_counter()
    tiempo_promedio = tiempo_promedio + tiempo_final - tiempo_inicial
```

Algoritmo de búsqueda lineal.

```
for reps in range(REPETICIONES):
    tiempo_inicial = time.perf_counter()
    r = interfaz_datos[0].shape[0]
    l = 0
    while r >= l:
        m = int((r + l) / 2)
        if interfaz_datos[0][m] == dato:
            break
        elif interfaz_datos[0][m] < dato:
            l = m + 1
        else:
            r = m - 1
    tiempo_final = time.perf_counter()
    tiempo_promedio = tiempo_promedio + tiempo_final - tiempo_inicial
```

Algoritmo de búsqueda binaria

Descripción de los experimentos

En mi caso, decidí cambiar los tamaños de la lista que analizar, debido a la poca resolución que tenían mis graficar a partir de valores superiores de 1000 elementos, ya que como se podrá apreciar, la diferencia de tiempo de ejecución entre ambas graficas es muy grande.

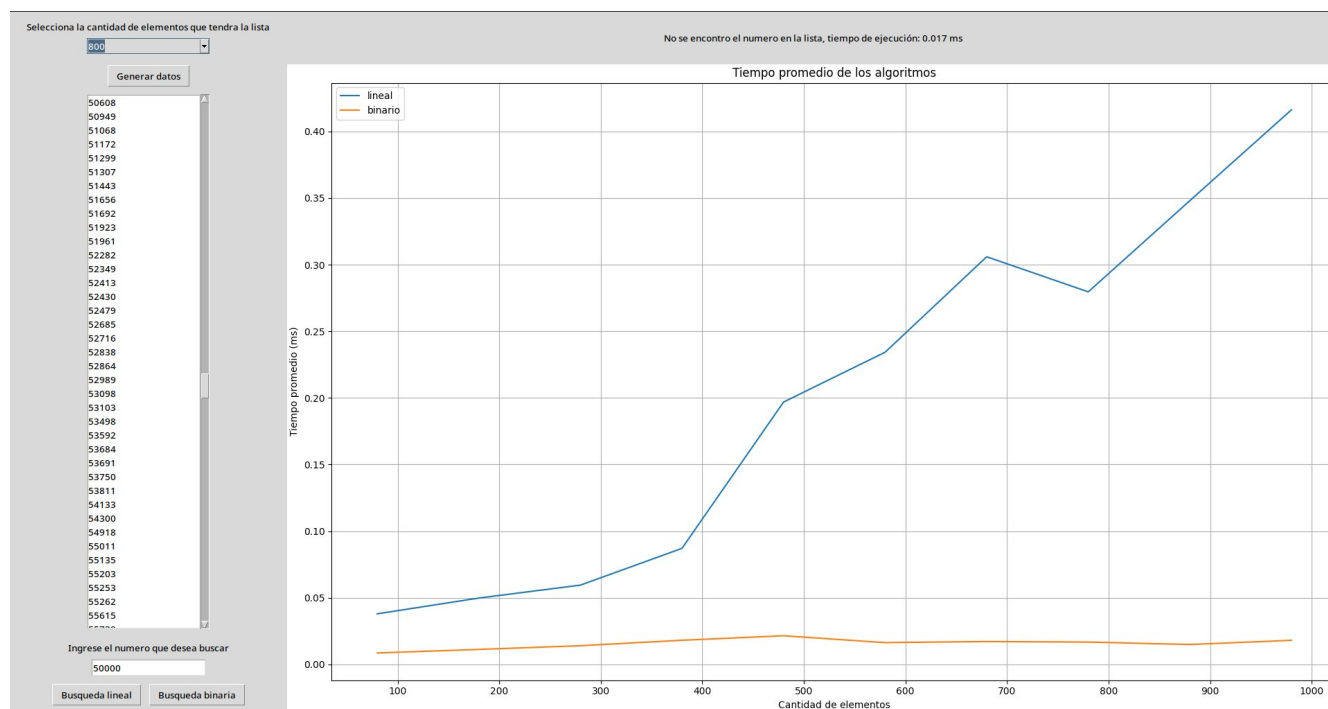
La cantidad de elementos de las listas que analicé a los largo de mi experimentación fueron valores de 100 a 1000 con un paso de 100, por lo que en total obtuve 10 tiempos distintos en milisegundos.

Con respecto a las repeticiones de cada uno de los algoritmos, puse un valor fijo de 30 repeticiones para cada una de las ejecuciones. Elegí un valor alto para eliminar variabilidad a la hora de obtener un promedio de tiempo, pero gracias a esto pude concluir que es

necesario un numero mas alto de repeticiones o que los sistemas computaciones son muy susceptibles al ruido.

Para los valores de búsqueda, generaba una lista aleatoria y se ordenaba. Durante todas las iteraciones de los algoritmos se buscaba un valor de 50000 en una lista cuyos valores eran aleatorios, pero se encontraban entre los limites de 0 a 100000. En todos los casos de la grafica de resultados, no se logró encontrar el valor de 50000 en las listas generadas. Esto quiere decir que para el algoritmo de búsqueda lineal siempre se generaba el peor caso posible.

Resultados



Como se puede apreciar en la grafica, tenemos una variación de tiempo considerable al algoritmo de búsqueda lineal. La diferencia entre el tiempo máximo y el tiempo mínimo es de aproximadamente 350 microsegundos, en comparación, el algoritmo de búsqueda binario tiene una variación despreciable, de aproximadamente 10 microsegundos.

Conclusiones

El algoritmo de búsqueda lineal es peor en todos los casos estudiados para buscar números en listas ordenadas frente a un algoritmo de búsqueda binario. Se puede realizar un mejor análisis si sumamos el tiempo de ejecución del algoritmo de ordenamiento a la hora de buscar un objeto dentro de la lista.

Repositorio de GITHUB:

<https://github.com/angel-lopez9177/analisis-algoritmos>