DATOS I
Adrian Pereira Ramírez
Jose Pablo Alpizar Mata

Problema 1

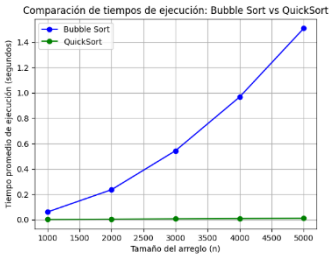
Pregunta 5

a. ¿Cuál algoritmo es más rápido y por qué?

QuickSort es más rápido que Bubble Sort. La razón principal es la diferencia en el Big O:

- QuickSort tiene una complejidad promedio de O(n log n), lo que lo hace mucho más eficiente en la mayoría de los casos.

- Bubble Sort tiene una complejidad de $O(n^2)$, lo que lo hace significativamente más lento, especialmente para arreglos grandes, como se observa en la gráfica.



A medida que el tamaño del arreglo aumenta, el tiempo de ejecución de Bubble Sort crece de manera cuadrática, mientras que QuickSort apenas aumenta.

b. ¿El tiempo de ejecución será el mismo si la implementación del algoritmo es iterativa o recursiva?

En términos de tiempo de ejecución, ambas versiones (iterativa y recursiva) deberían ser igual de rápidas si están bien implementadas. Sin embargo, la versión iterativa es mejor en sistemas con poca memoria, porque no usa espacio adicional en la pila como lo hace la versión recursiva.

c. ¿Es posible que exista un algoritmo de ordenamiento que sea muy eficiente en consumo de recursos pero que a la vez sea relativamente rápido?

Sí, hay algoritmos como Merge Sort que son rápidos en términos de tiempo de ejecución y en algunos casos, utilizan menos memoria. Por ejemplo, QuickSort es muy rápido, pero al ser recursivo puede consumir más memoria. Merge Sort, aunque usa más espacio adicional para combinar los subarreglos, es más predecible en cuanto al uso de memoria.

d. Suponga que se planea ejecutar el algoritmo en un sistema computacional con extremadamente bajos recursos de memoria. ¿Cuál de los dos algoritmos de ordenamiento escogería y por qué?

En un sistema con extremadamente bajos recursos de memoria, sería preferible usar Bubble Sort, ya que es un algoritmo en el que se realiza el intercambio de elementos directamente en

el mismo arreglo sin necesidad de espacio adicional. Aunque es mucho más lento, consume menos memoria que QuickSort, que puede requerir espacio en la pila para las llamadas recursivas, lo que es malo en sistemas con poca memoria.

PROBLEMA 2

1. ¿Cuál es la diferencia entre el algoritmo de búsqueda lineal y búsqueda por interpolación?

R/=La búsqueda lineal, en este algoritmo, se busca un elemento atravesando todo el array y comparando cada elemento con el elemento deseado para encontrar una coincidencia. Es simple y funciona en listas desordenadas, y listas cortas. Mientras que, la búsqueda por interpolación es un algoritmo de búsqueda rápido y eficiente. Mejora el algoritmo de búsqueda binaria para escenarios donde los elementos del array se distribuyen uniformemente sobre el array ordenada. Además, es mucho más eficiente en listas grandes.

2. Suponga que se tiene que buscar un elemento en una lista desordenada, pero se desea optimizar el tiempo de búsqueda por sobre cualquier otra métrica ¿Cómo se podría hacer eso?

R/= Tomando en cuenta que es una lista desordenada, lo preferible seria, en primer lugar, ordenarla debido a que esto permite implementar técnicas de búsqueda mas eficientes como la búsqueda por interpolación. Para ordenarla se debe usar un algoritmo de ordenamiento, ya sea QuickSort, MergeSort y entre otros. Cuando ya está ordenada la lista, empleamos un algoritmo de búsqueda como la búsqueda por interpolación, lo que permite buscar el elemento que se ocupa de manera rápida y eficaz. Aunque, esto en caso de la lista sea grande. Si la lista es pequeña, se puede usar búsqueda lineal, así no se ocupa ordenar la lista, haciendo el proceso más rápido y sigue siendo eficiente, pero solo si la lista es corta.

3. Busque y explique alguna aplicación de la vida real donde el tiempo de búsqueda en una lista o en un arreglo sea crítico para que la aplicación se pueda dar.

Cuando se usan aplicaciones como Netflix, Amazon o Spotify, estos necesitan buscar y procesar rápidamente grandes volúmenes de datos para ofrecer contenido que los usuarios quieren ver o sugerencias basadas en las preferencias de los usuarios. Entonces los algoritmos revisan el historial de búsquedas, compras, o reproducciones anteriores para identificar patrones y preferencias. El tiempo es crítico por que los usuarios esperaran resultados rápidos y no esperar mucho. Por otra parte, el algoritmo tiene que actualizarse en tiempo real constantemente por que los usuarios cambian de preferencias o realizan diferentes búsquedas la mayoría del tiempo, por tanto la actualización debe ser rápida y constante.