

Integrantes:

Esteban Campo Abarca

Deiler Morera Valverde

Problema 1:

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

Quicksort es generalmente más rápido que Bubble Sort porque tiene una complejidad de tiempo promedio de $O(n \log n)$, mientras que Bubble Sort tiene $O(n^2)$.

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

No necesariamente. La implementación recursiva puede ser más lenta debido a la sobrecarga de la pila de llamadas. La versión iterativa puede ser más eficiente en tiempo y espacio.

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, Merge Sort y Heapsort son ejemplos de algoritmos que son eficientes en tiempo ($O(n \log n)$) y consumen recursos razonablemente bien.

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?

Elegiría Quicksort porque es un algoritmo in-place, usando poca memoria adicional y siendo eficiente en tiempo de ejecución.

Problema 2:

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

Búsqueda Lineal: Busca elemento por elemento. Complejidad $O(n)$. Funciona en listas desordenadas y ordenadas.

Búsqueda por Interpolación: Usa la posición del valor esperado para buscar. Complejidad $O(\log \log n)$ en promedio. Solo funciona en listas ordenadas.

2. ¿Cómo se podría optimizar el tiempo de búsqueda en una lista desordenada?

Usar una tabla hash. Permite buscar en $O(1)$ en promedio, aunque requiere más memoria.

3. ¿Ejemplo de aplicación donde el tiempo de búsqueda es crítico?

Sistemas de búsqueda de información en bases de datos. Por ejemplo, reservas de vuelos. Un tiempo de búsqueda rápido es crucial para la satisfacción del usuario y el rendimiento del sistema.

Graficas:



