**Análisis de Búsqueda Binaria y Algoritmos de Ordenamiento**

**Parte 1: Búsqueda Binaria**

Se trabajó con una lista desordenada de libros, cada uno con un código como 'L001'. Primero, se aplicó la búsqueda binaria directamente sobre esta lista sin ordenarla.

**Resultado sin ordenar**

La búsqueda binaria falló o devolvió un resultado incorrecto, aunque el libro 'L001' estaba presente en la lista.

**Pregunta de análisis posterior**

¿Qué resultados inesperados observaste? ¿Por qué ocurre esto si el libro sí está en la lista?

La búsqueda binaria no encontró el libro. Esto ocurre porque el algoritmo de búsqueda binaria requiere que la lista esté ordenada previamente para funcionar correctamente. Cuando la lista está desordenada, el algoritmo puede descartar la parte de la lista donde se encuentra el libro.

**Resultado con lista ordenada**

Luego de ordenar la lista por el campo 'código', la búsqueda binaria encontró correctamente el libro 'L001'.

**Pregunta de análisis posterior (después de ordenar)**

¿Qué diferencias observaste entre los dos resultados? ¿Qué nos dice esto sobre las condiciones necesarias para aplicar correctamente la búsqueda binaria?

La principal diferencia fue que, sin ordenar, el algoritmo falló; mientras que, con la lista ordenada, la búsqueda binaria encontró el libro exitosamente. Esto confirma que una condición indispensable para aplicar búsqueda binaria es que la lista esté ordenada.

**Parte 2: Ordenamiento de Lista Casi Ordenada**

Se usó una lista casi ordenada de libros (solo un elemento fuera de lugar) para comparar el comportamiento de dos algoritmos de ordenamiento: Burbuja e Inserción. Ambos algoritmos ordenaron la lista correctamente, pero con diferentes cantidades de comparaciones e intercambios.

**Resultados del algoritmo Burbuja:**

Comparaciones realizadas: 15

Intercambios realizados: 1

Observación: Aunque solo un elemento estaba desordenado, Burbuja realizó muchas comparaciones innecesarias.

**Resultados del algoritmo Inserción:**

Comparaciones realizadas: 6

Intercambios realizados: 1

Observación: Inserción detectó rápidamente el desorden mínimo y ajustó solo lo necesario.

**Pregunta de análisis posterior:**

¿Qué algoritmo de ordenamiento es más eficiente en una lista casi ordenada: burbuja o inserción? ¿Por qué?

El algoritmo de inserción es más eficiente en listas casi ordenadas. Esto se debe a que realiza menos comparaciones e intercambios, ya que detecta rápidamente que la mayoría de los elementos ya están en el orden correcto. Por el contrario, el algoritmo burbuja sigue recorriendo toda la lista y realizando comparaciones innecesarias, lo que lo hace menos eficiente en este tipo de casos.