

Una vez terminada la comparación responde las siguientes preguntas:

1. ¿Cuál de los métodos de búsqueda, en su opinión, resultó ser el más eficiente y por qué?

El quinto metodo y creo que es por usar un puntero a puntero

2. ¿Qué diferencias observaron al comparar los tiempos de ejecución de los métodos de búsqueda?

Que cuando se utilizan punteros, el tiempo no varia tanto

3. ¿Qué creen que podría estar influyendo en la diferencia de tiempo de impresión entre los métodos? ¿Cómo podrían optimizar la velocidad de impresión?

La optimizacion y en donde se encuentren las funciones

4. ¿Cuál es la ventaja de utilizar punteros y fórmulas de direccionamiento en comparación con la manipulación tradicional de arrays? ¿Pueden citar ejemplos específicos donde esto resultó beneficioso?

La velocidad, pero sacrifica un poco la legibilidad

5. ¿Pueden identificar situaciones en las que el uso de punteros y fórmulas de direccionamiento podría no ser la mejor opción? ¿En qué casos la manipulación tradicional de arrays sería preferible?

En lo personal prefiero la tradicional, pues a pesar de que entregas un poco de rendimiento, optienes legibilidad, y mejor aun cuando le tienes que dar soporte

6. ¿Cómo afecta la complejidad del código y la legibilidad al elegir entre métodos que utilizan punteros y fórmulas de direccionamiento y métodos más convencionales? ¿Existen situaciones en las que la claridad del código es más importante que la eficiencia?

Que tienes que saber si apunta a la direccion al valor, si existen muchas situaciones

7. ¿Pueden proporcionar ejemplos de casos de uso en los que el rendimiento (tiempo de ejecución) es un factor crítico y, por lo tanto, los métodos con punteros y fórmulas de direccionamiento son esenciales?

Cuando se trata de circuitos pequenos o casos en los que necesites que sea lo mas rapido posible