



Taller 8. Experimentación con Punteros y Arrays.

Martes 31 de octubre del 2023

Hernandez Cesena Ivan Fernando 373077 932

Instrucciones: Durante este taller, vamos a resolver ejercicios relacionados con la unidad 5. Para cada ejercicio, declaren las variables, constantes y funciones necesarias para llevar a cabo la tarea requerida. Realicen el código a mano de los siguientes ejercicios*.

1. **Replicar Códigos:** Comienza reproduciendo los códigos correspondientes a las cinco formas de trabajo con punteros y arrays presentadas en clase.
2. **Implementar Función de Búsqueda Lineal:** Desarrolla una función que permita realizar una búsqueda lineal para encontrar uno o varios números o palabras en los datos generados. Asegúrate de que el nombre de esta función siga la nomenclatura establecida, por ejemplo, `busquedaLineal_Metodo1()`.
3. **Comparación de Métodos:** Luego, adapta los códigos para realizar una comparación exhaustiva de todos los métodos. En este proceso, se generará un conjunto de 'n' números aleatorios, seguido de 'n' números de búsqueda. Esto permitirá iterar las búsquedas según la cantidad de veces indicada, utilizando nuevos datos y búsquedas en cada iteración.
4. **Medición del Tiempo de Ejecución:** Al final, asegúrate de obtener el tiempo de ejecución total para cada método en todas las búsquedas realizadas. Esto te proporcionará información valiosa sobre la eficiencia de cada enfoque. Además, también mide el tiempo de impresión de los resultados para cada método.

Nota: Recuerda seguir la nomenclatura para las funciones de búsqueda e impresión, lo que facilitará la identificación y organización de cada método, permitiéndote realizar una comparación y un análisis efectivos de los resultados. No olvides documentar el código para futuras referencias.

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 método 2 que usa un vector. Creo que es el más rápido por la fórmula que usa y porque a diferencia de los demás este usa un vector no una matriz..

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

muy poca. para iteraciones de 3 cifras.



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

usando punteros a punteros para las ubicaciones.

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?

es mas rapido buscar en un espacio dónde estás seguro de que vas a encontrar lo que buscas que a buscar en todos los registros de almacenamiento uno por uno.

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?

para arreglos de no gran tamaño.

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?

se vuelve un poco más complicado de entender pero a fin de cuentas no es una fórmula muy grande, solo sería cosa de acostumbrarse a usarlo.

Las situaciones en las que en tu código importa mucho la claridad es cuando estás haciendo un código muy grande y en este caso no creo que se vea afectada la claridad del código, porque todo está separado por funciones, eso lo vuelve más claro.

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 tienen registros muy grandes es muy importante el tiempo de ejecución, ejemplo:

tenemos una base de datos de 10,000 usuarios y se busca un solo usuario, esos 10,000 datos están en varios registros de la memoria computadora y sería mas tardado buscando registro por registro en la memoria a ya tener la ubicación de donde están los datos de los usuarios y desde ahí encontrar al usuario pedido.