

M1 Programación I FIN A

Proyecto Integrador

|  |  |
| --- | --- |
| **Tutor:** | **Adriana Hernandez Garcia** |
| **Estudiante:** | **José Ramón Ibáñez Posadas** |
| **Matricula:** | **BNL098377** |

|  |  |
| --- | --- |
| Monterrey, Nuevo León | Sábado, 03 de Febrero de 2024 |

Introducción

La búsqueda en arreglos y los métodos de ordenamiento son conceptos fundamentales en la ciencia de la computación y la programación. La búsqueda en arreglos implica encontrar un valor específico dentro de una colección de elementos almacenados en un arreglo, mientras que los métodos de ordenamiento se utilizan para organizar los elementos de un arreglo en un orden determinado. Estos temas son esenciales para comprender cómo manipular y trabajar eficientemente con conjuntos de datos en la programación.

**Búsqueda en arreglos:** La búsqueda en arreglos es un proceso mediante el cual se busca un valor específico dentro de una colección de elementos almacenados en un arreglo. Se pueden utilizar diversos algoritmos de búsqueda, como la búsqueda lineal y la búsqueda binaria. La búsqueda binaria es especialmente eficiente cuando el arreglo está ordenado, ya que reduce el espacio de búsqueda a la mitad en cada iteración. En general, la búsqueda en arreglos es una operación esencial para encontrar información dentro de conjuntos de datos estructurados.

**Cuándo se puede utilizar una búsqueda binaria:** La búsqueda binaria se puede utilizar cuando el arreglo está ordenado, ya sea en orden ascendente o descendente. Este algoritmo es eficiente y rápido, especialmente para arreglos grandes, ya que reduce el espacio de búsqueda a la mitad en cada iteración. La búsqueda binaria es útil para encontrar un elemento específico en un arreglo ordenado de manera eficiente, especialmente en situaciones donde se realizan múltiples búsquedas o cuando se necesita un tiempo de respuesta rápido.

**Métodos de ordenamiento:** Existen varios métodos de ordenamiento que se utilizan para organizar los elementos de un arreglo en un orden determinado. Algunos de los métodos más comunes incluyen el ordenamiento burbuja, el ordenamiento por inserción, el ordenamiento por selección, el ordenamiento por fusión, el ordenamiento rápido y el ordenamiento por radix. Cada método tiene sus propias características y complejidades, y la elección del método adecuado depende del tamaño del arreglo, la distribución de los datos y los requisitos de rendimiento del sistema. Los métodos de ordenamiento son esenciales para realizar operaciones de búsqueda eficientes y para optimizar el procesamiento de datos en la programación.

Desarrollo

Cuestionario

1. **¿Qué es la búsqueda en arreglos?**

La búsqueda en arreglos es un proceso mediante el cual se busca un valor específico dentro de una colección de elementos almacenados en un arreglo. Este proceso implica recorrer el arreglo y comparar cada elemento con el valor buscado hasta encontrar una coincidencia o llegar al final del arreglo sin éxito.

Existen varios algoritmos de búsqueda que se utilizan dependiendo de diversos factores, como el tamaño del arreglo, si el arreglo está ordenado o no, y el tiempo de ejecución requerido. Algunos de los algoritmos de búsqueda más comunes son:

**Búsqueda lineal (secuencial)**: Recorre el arreglo secuencialmente desde el principio hasta el final, comparando cada elemento con el valor buscado. Es simple pero puede ser ineficiente para arreglos grandes.

**Búsqueda binaria**: Solo se puede aplicar en arreglos ordenados. Este algoritmo divide repetidamente el arreglo en mitades y descarta una mitad basándose en si el valor buscado es menor o mayor que el elemento central. Es muy eficiente para arreglos ordenados, ya que reduce el espacio de búsqueda a la mitad en cada iteración.

**Búsqueda por interpolación**: Similar a la búsqueda binaria, pero en lugar de dividir el arreglo en partes iguales, este algoritmo calcula una estimación de la posición del valor buscado basándose en el rango de valores y la posición de los elementos en el arreglo.

**Búsqueda por hashing**: Utiliza una función de hash para mapear claves de búsqueda a índices en una estructura de datos como una tabla hash. Esta búsqueda es muy eficiente en promedio, pero requiere una buena función de hash y puede tener colisiones.

La elección del algoritmo de búsqueda adecuado depende de factores como el tamaño del arreglo, la cantidad de búsquedas que se realizarán y si el arreglo está ordenado. Cada algoritmo tiene sus ventajas y desventajas en términos de eficiencia y complejidad.

1. **¿Cuándo se puede utilizar una búsqueda binaria?**

La búsqueda binaria es un algoritmo de búsqueda eficiente que se puede utilizar cuando el arreglo está ordenado. Este algoritmo se basa en la división repetida del arreglo en mitades y descarta una mitad basándose en si el valor buscado es menor o mayor que el elemento central.

Aquí hay algunas situaciones en las que se puede utilizar la búsqueda binaria de manera efectiva:

**Arreglos ordenados:** La búsqueda binaria solo funciona en arreglos ordenados, ya sea en orden ascendente o descendente. Si el arreglo no está ordenado, se debe ordenar primero antes de aplicar la búsqueda binaria, lo cual puede llevar tiempo adicional.

**Grandes arreglos:** La búsqueda binaria es especialmente útil en arreglos grandes, ya que reduce el espacio de búsqueda a la mitad en cada iteración. Esto significa que su tiempo de ejecución es O(log n), donde n es el número de elementos en el arreglo. En comparación, la búsqueda lineal tiene un tiempo de ejecución de O(n), donde n es el número de elementos en el arreglo.

**Búsqueda de elementos específicos:** La búsqueda binaria es ideal para encontrar un elemento específico en un arreglo ordenado. Si solo necesitas verificar si un elemento está presente en el arreglo, la búsqueda binaria puede ser más eficiente que la búsqueda lineal, especialmente en arreglos grandes.

**Repetidas búsquedas:** Si planeas realizar múltiples búsquedas en el mismo arreglo, puede ser beneficioso ordenarlo una vez y luego realizar búsquedas binarias repetidas. Una vez que el arreglo está ordenado, la búsqueda binaria proporciona un tiempo de búsqueda constante en cada consulta.

1. **Explica cada uno de los métodos de ordenamiento.**

**Ordenamiento Burbuja (Bubble Sort):**

Este método compara repetidamente pares de elementos adyacentes y los intercambia si están en el orden incorrecto.

Es un algoritmo simple y fácil de implementar, pero no es eficiente para arreglos grandes debido a su complejidad temporal de O(n^2).

**Ordenamiento por Inserción (Insertion Sort):**

En este método, se considera una parte del arreglo como ordenada y el resto como no ordenada. Se toma cada elemento de la parte no ordenada y se inserta en la posición correcta en la parte ordenada.

Es eficiente para arreglos pequeños o casi ordenados, pero también tiene una complejidad temporal de O(n^2) en el peor de los casos.

**Ordenamiento por Selección (Selection Sort):**

El algoritmo de selección busca repetidamente el elemento mínimo de la parte no ordenada del arreglo y lo intercambia con el primer elemento no ordenado.

Aunque es simple y fácil de implementar, su complejidad temporal es O(n^2) en todos los casos.

**Ordenamiento por Fusión (Merge Sort):**

Este método sigue el enfoque de divide y conquista. Divide repetidamente el arreglo en mitades hasta que cada subarreglo tenga un solo elemento, luego fusiona los subarreglos en orden.

Tiene una complejidad temporal de O(n log n) en todos los casos, lo que lo hace más eficiente que los métodos anteriores para arreglos grandes.

**Ordenamiento Rápido (Quick Sort):**

Al igual que el merge sort, el quick sort también sigue el enfoque de divide y conquista. Se elige un elemento pivote y se rearrange el arreglo de modo que los elementos menores al pivote estén a su izquierda y los mayores a su derecha.

Tiene una complejidad temporal de O(n log n) en promedio, pero en el peor caso puede llegar a O(n^2) si el pivote es elegido de forma subóptima.

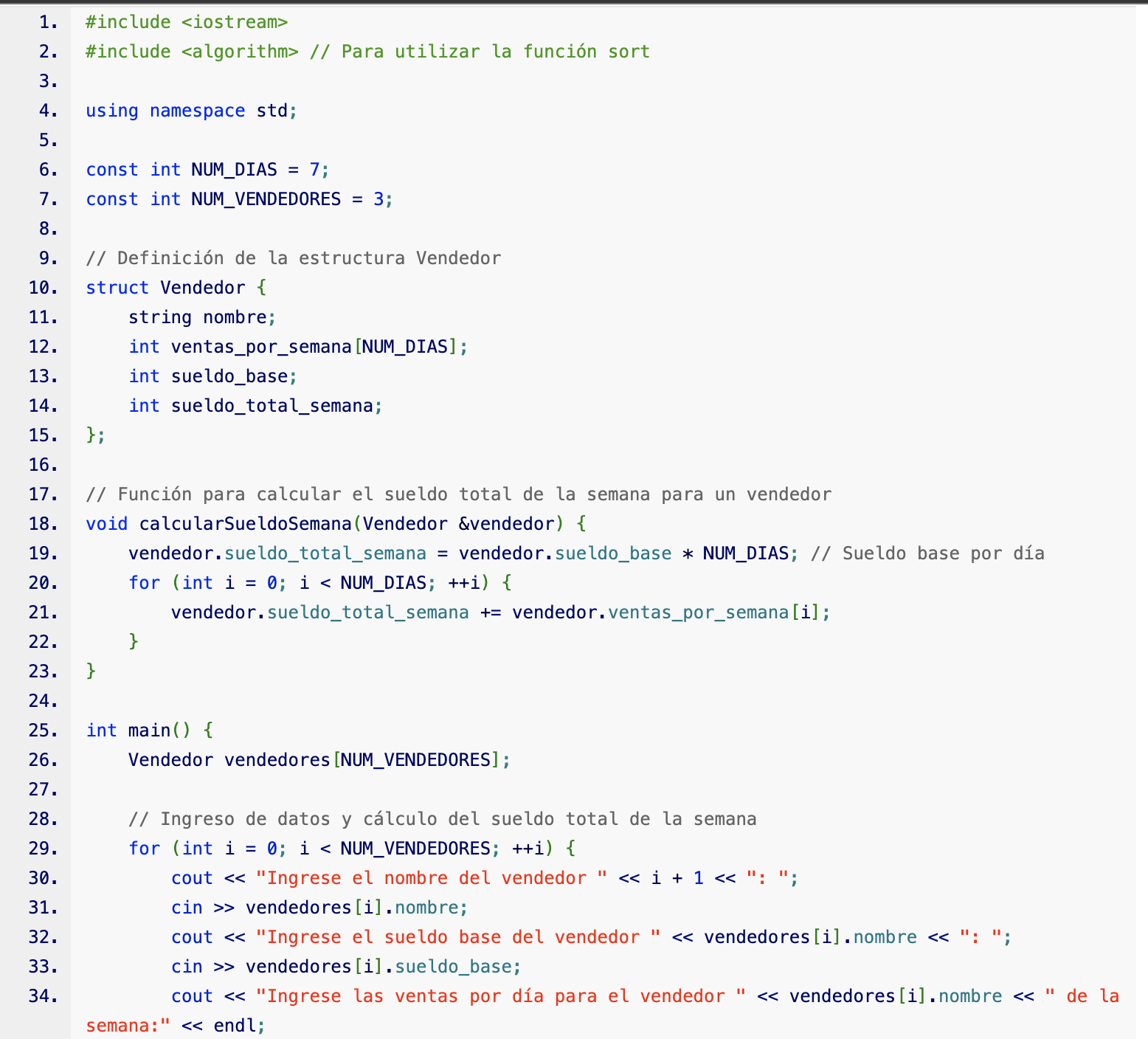
**Ordenamiento por Radix (Radix Sort):**

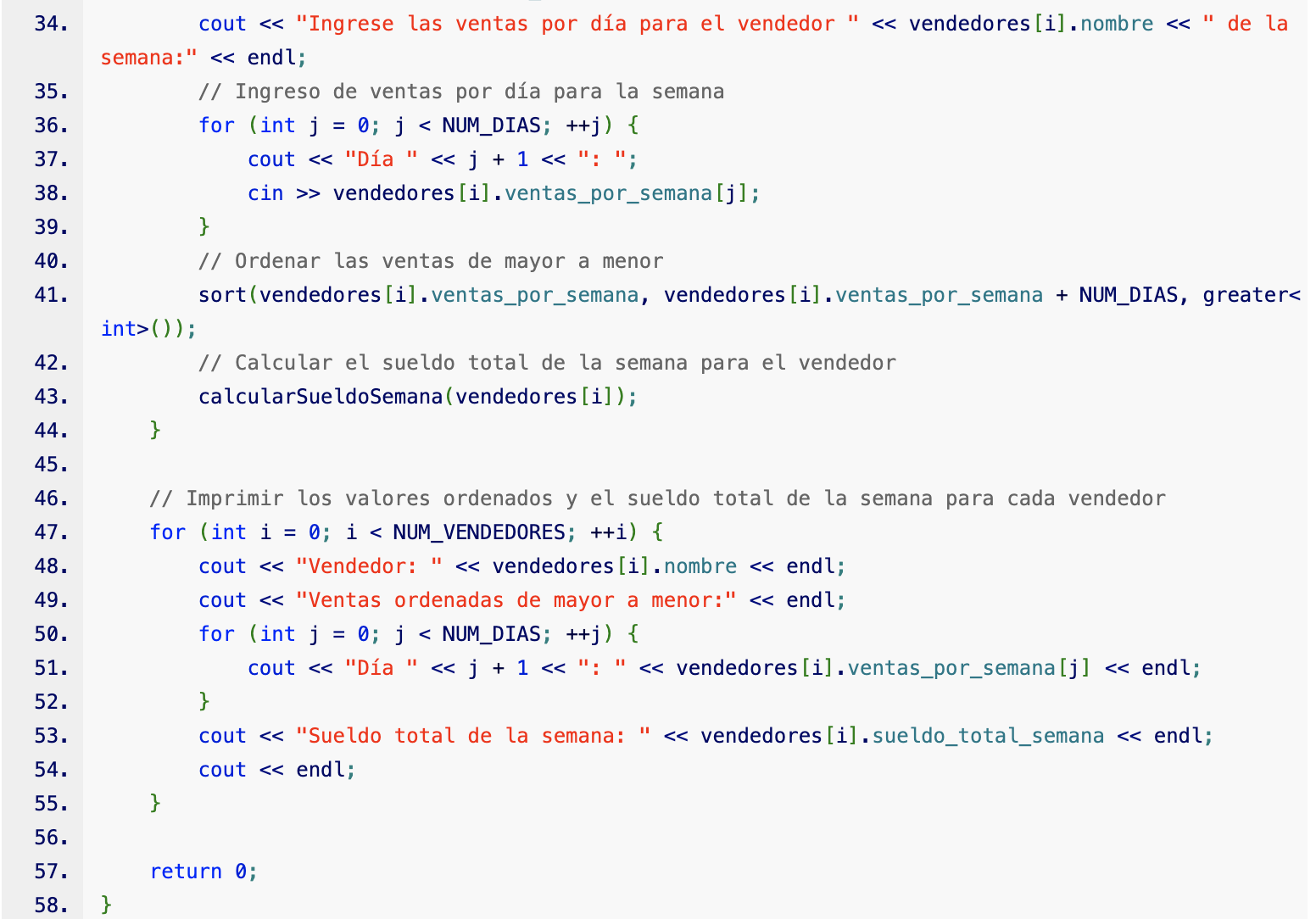
Este método clasifica los elementos basándose en los dígitos o letras individuales que componen su valor. Se ordenan primero los dígitos menos significativos y luego los más significativos.

Es eficiente para ordenar enteros o cadenas de caracteres con una longitud fija, y tiene una complejidad temporal de O(kn), donde k es el número de dígitos o letras.

1. **Deberás modificar el código que realizaste en el Proyecto Modular en el deberás anexar lo siguiente y explicar las modificaciones que realizaste, anexar el código, así como capturas de pantalla.** 
   1. **Deberás anexar un arreglo en donde guardes las ventas por día por semana para cada vendedor, deberá obtener el sueldo total de la semana.**
   2. **Al ingresar los valores de las ventas deberás ordenar de mayor a menor, puede utilizar el método de ordenamiento que prefieras.**
   3. **Los valores ordenados y el sueldo total de la semana se deberán imprimir.**

Capturas de Pantalla







Liga Código

<https://ideone.com/zDqII1>

Liga del Vídeo

<https://youtu.be/wcIttEZ_OK4>

Conclusión

La búsqueda en arreglos y los métodos de ordenamiento son conceptos fundamentales en la programación y la ciencia de la computación. La búsqueda en arreglos nos permite encontrar elementos específicos dentro de una colección de datos, mientras que los métodos de ordenamiento nos permiten organizar esos datos de manera eficiente en un orden determinado.

La búsqueda binaria destaca como un algoritmo eficiente cuando el arreglo está ordenado, ya que reduce el espacio de búsqueda de manera significativa en cada iteración. Esto lo convierte en una opción ideal para arreglos grandes y para situaciones donde se requiere una respuesta rápida.

Por otro lado, los métodos de ordenamiento son esenciales para mantener los datos organizados y facilitar operaciones como la búsqueda y la recuperación de información. Cada método de ordenamiento tiene sus propias ventajas y desventajas, y la elección del método adecuado depende del tamaño del conjunto de datos, la distribución de los elementos y los requisitos de rendimiento del sistema.

En resumen, la comprensión de la búsqueda en arreglos y los métodos de ordenamiento es fundamental para cualquier programador, ya que proporciona las herramientas necesarias para trabajar eficientemente con conjuntos de datos y realizar operaciones de búsqueda y ordenamiento de manera efectiva en la programación.

# **Bibliografía**

Univesidad Virtual CNCI. (2024). Tema 14 Algoritmos de Busqueda. 2024, Recuperado de: <https://cnci.blackboard.com/bbcswebdav/courses/BbCont10/Contenido/Ingenierias/I_ProgramacionI_OCT19/m04-t14/m04-t14.html>

Univesidad Virtual CNCI. (2024). Tema 15 Algoritmos de Ordenamiento. 2024, Recuperado de: <https://cnci.blackboard.com/bbcswebdav/courses/BbCont10/Contenido/Ingenierias/I_ProgramacionI_OCT19/m04-t15/m04-t15.html>