



Arturo Sánchez Rodríguez | A01275427

21 de Abril del 2023

Programación de estructuras de datos y algoritmos fundamentales (Gpo 850)

Profesor: Dr. Eduardo Arturo Rodríguez Tello

Act-Integradora-2 Estructuras de Datos Lineales

Introducción

Este trabajo trata sobre el desarrollo de una aplicación en equipo que cumpla con diferentes objetivos, relacionados con el manejo y organización de datos en una bitácora. El trabajo consiste en leer un archivo de entrada llamado "bitacora.txt", almacenar los datos en una Doubly Linked List, ordenar la información por fecha y hora, y permitir la búsqueda de información dentro de un rango de fechas específico.

Para llevar a cabo estos objetivos, se deben implementar diferentes algoritmos y estructuras de datos. Se utilizará una Doubly Linked List para almacenar y ordenar los datos, y se implementarán los algoritmos de Merge Sort y QuickSort para comparar su desempeño en el proceso de ordenamiento. Además, se utilizará el algoritmo de búsqueda binaria para encontrar las fechas de inicio y fin de la búsqueda de información.

En el proceso de desarrollo, los integrantes del equipo deberán reflexionar sobre la importancia y eficiencia del uso de diferentes estructuras de datos lineales en situaciones de este tipo, y analizar la complejidad computacional de las operaciones básicas de la estructura de datos empleada en la implementación. También deberán analizar los resultados de la comparación de los dos algoritmos de ordenamiento desarrollados, contrastando su desempeño con la complejidad temporal de los mismos.

El trabajo culminará con la generación de un documento llamado "ReflexActInt-2.pdf", en el que se detallarán las reflexiones individuales de cada integrante del equipo, así como los resultados y conclusiones del proyecto en general. Se deberán incluir referencias bibliográficas en el documento para respaldar las afirmaciones y conclusiones.

Reflexión

En general, la elección de una estructura de datos lineal adecuada para una situación problema depende de las características específicas de la tarea a resolver. En el caso de este reto, la implementación de un sistema de reservas de asientos en un teatro requiere que se puedan insertar y eliminar asientos de manera eficiente, además de realizar búsquedas de asientos disponibles para reservar.

Una Linked List (lista enlazada) es una estructura de datos en la que cada elemento (nodo) almacena un valor y un puntero al siguiente elemento de la lista. La inserción y eliminación de elementos en una Linked List es eficiente, ya que solo se requiere actualizar los punteros de los nodos adyacentes. Sin embargo, para buscar un elemento específico en una Linked List es necesario recorrer la lista desde el principio, lo que tiene una complejidad de $O(n)$, siendo n el número de elementos en la lista.

Una Doubly Linked List (lista doblemente enlazada) es una estructura de datos en la que cada nodo tiene un puntero al nodo anterior y al siguiente. Esto permite recorrer la lista en ambas direcciones, lo que es útil para buscar elementos y también para eliminarlos de manera eficiente. La complejidad de búsqueda en una Doubly Linked List es $O(n)$, al igual que en una Linked List, pero la eliminación de un elemento puede ser más rápida ya que no es necesario recorrer la lista desde el principio.

En el caso de este reto, se prefiere utilizar una Doubly Linked List en lugar de una Linked List debido a que se necesitan realizar búsquedas frecuentes de asientos disponibles y eliminar asientos reservados. Además, la implementación de un sistema de reservas de asientos en un teatro requiere que se pueda realizar un recorrido bidireccional por los asientos para verificar la disponibilidad de asientos contiguos.

La complejidad de inserción y eliminación en una Doubly Linked List es $O(1)$ en el peor caso, ya que solo es necesario actualizar los punteros de los nodos adyacentes. La complejidad de búsqueda en una Doubly Linked List es $O(n)$, pero en la práctica se pueden implementar diversas estrategias para optimizar esta operación, como por ejemplo mantener una referencia al nodo más reciente buscado y comenzar la búsqueda a partir de ahí.

Act-Integradora-2 Estructuras de Datos Lineales

En cuanto a los algoritmos de ordenamiento, se desarrollaron dos implementaciones: una basada en Bubble Sort y otra en Merge Sort. La complejidad temporal en el peor caso de Bubble Sort es $O(n^2)$, mientras que la complejidad temporal en el peor caso de Merge Sort es $O(n \log n)$. En la práctica, Merge Sort es significativamente más rápido que Bubble Sort para listas largas, ya que tiene una tasa de crecimiento mucho más baja de la complejidad temporal a medida que aumenta el tamaño de la lista.

En conclusión, la elección de una estructura de datos lineal adecuada es esencial para lograr una solución eficiente a un problema específico. En el caso de este reto, la Doubly Linked List es preferible a la Linked List debido a su capacidad para realizar búsquedas y eliminaciones eficientes.

Act-Integradora-2 Estructuras de Datos Lineales

GitHub.-

Referencias

- Fuentes, J. (2023, 17 abril). Qué son las estructuras de datos y por qué son tan útiles. *OpenWebinars.net*.
<https://openwebinars.net/blog/que-son-las-estructuras-de-datos-y-por-que-son-tan-utiles/#:~:text=Las%20estructuras%20de%20datos%20son%20%C3%BAtiles%20porque%20nos%20permiten%20tener,que%20trabajan%20con%20Big%20Data.>
- Henry, R. (2022). ¿Qué es una estructura de datos en programación? | Henry. *Henry*.
<https://blog.soyhenry.com/que-es-una-estructura-de-datos-en-programacion/>