



Informe Tarea 0

27 de abril de 2024
Manuel Espinoza Quintero
20642598

Motivación

Las estructuras de datos como heaps y árboles son clave para la búsqueda óptima de datos por su organización eficiente. Los heaps facilitan acceso rápido a elementos prioritarios, mientras que los árboles permiten operaciones rápidas de búsqueda, inserción y eliminación, crucial para manejar grandes volúmenes de datos eficientemente. Para esta tarea era de gran importancia ocupar estas estructuras para el rendimiento del programa.

Informe

Estructuras de Datos Usadas y sus Características

Parte 1: Heap Market

Estructura de Datos: Se utilizan estructuras tipo heap para manejar las órdenes de compra y venta.

Complejidad de Operaciones:

- *Insertar (pushCompra y pushVenta):* $O(\log n)$ debido a la necesidad de mantener el orden del heap.
- *Eliminar (popCompra y popVenta):* $O(\log n)$ ya que, al remover el elemento superior, es necesario reorganizar el heap.

Parte 2: A buscar cabezas

- **Head:** Almacena datos de cada cabeza con atributos como ID, nombre, año, región, y coordenadas. Usado para el manejo de datos individuales.
- **TreeNode y YearRegionNode:** Nodos de árboles binarios de búsqueda (BST), donde *TreeNode* es para búsquedas por ID y año, y *YearRegionNode* es específico para búsquedas combinadas de año y región. Estos nodos permiten operaciones de búsqueda, inserción y eliminación con una complejidad promedio de $O(\log n)$, adecuada para manejar grandes volúmenes de datos de manera eficiente.

Implementación en Memoria de las Estructuras Usadas

Parte 1: Heap Market

Las órdenes de compra y venta se almacenan en arrays de tipo `Orden` que funcionan como heaps, uno como max-heap para las compras y otro como min-heap para las ventas. Estos arrays permiten realizar inserciones y extracciones rápidas manteniendo el orden deseado.

Parte 2: A buscar cabezas

En memoria, los nodos de los árboles (*TreeNode* y *YearRegionNode*) se conectan mediante punteros a otros nodos a la izquierda o derecha, formando una estructura jerárquica que facilita la búsqueda rápida. Los datos de las cabezas se enlazan directamente a estos nodos, permitiendo accesos rápidos y evitando duplicación de datos.

Complejidad de las Operaciones en el Heap Market

Todas las operaciones principales (comprar, vender, procesar estado) se diseñan para trabajar con una complejidad de $O(\log n)$ respecto a la cantidad de órdenes activas.

Consideraciones y Estrategias en los Árboles de Búsqueda

Se utilizan diferentes tipos de árboles para diferentes tipos de búsquedas: un árbol por ID, otro por año, y un tercero por año y región. Esto permite que las búsquedas sean más eficientes al segmentar los datos según el tipo de consulta, asegurando que cada árbol maneje solo los datos pertinentes a su criterio de búsqueda.

- Árbol Binario de Búsqueda (BST): Empleado para organizar las cabezas por ID y por año, permitiendo búsquedas eficientes con una complejidad de $O(\log n)$.
- Árbol AVL: Utilizado para búsquedas por año y región, este tipo de árbol se mantiene balanceado automáticamente para garantizar operaciones rápidas y eficientes también en $O(\log n)$.

Otras Aplicaciones de las Estructuras de Datos Usadas

Parte 1: Heap Market

Los heaps son ampliamente utilizados para mantener un conjunto de elementos ordenados, facilitando operaciones como la obtención del mínimo o máximo en tiempo constante. Son cruciales en algoritmos de ordenamiento como heapsort, en la implementación de colas de prioridad y en sistemas de planificación de procesos en sistemas operativos. Por ejemplo, en alguna cola de prioridad de venta de entradas para una pagina con mucho flujo.

Parte 2: A buscar cabezas

Los árboles binarios de búsqueda se utilizan en bases de datos y sistemas de archivos para permitir búsquedas rápidas y manejo eficiente de grandes cantidades de datos. Son especialmente útiles en aplicaciones donde se requiere inserción, eliminación y acceso rápido, como sistemas de gestión de inventario o en la implementación de índices en sistemas de bases de datos relacionales.

Conclusión

El uso de estructuras de datos como heaps y árboles binarios de búsqueda sirven para gestionar eficientemente operaciones, para la tarea específicamente: transacciones en el "Heap Marketz manejo de datos en .^ buscar cabezas". Las estructuras implementadas permiten operaciones rápidas con complejidad $O(\log n)$. Finalmente se puede analizar el uso real de esto en sistemas de manejo de órdenes y datos para asegurar eficiencia.