Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2133 - Estructura de Datos y Algoritmos

Informe Tarea 1

24 de abril de 2024
Monserrat Diaz
21626545

Motivación

En esta tarea se buscaba aplicar Heaps y Árboles de Búsqueda para resolver un problema dado. Para el caso de Heap Market se utilizaron heaps, donde el heap de compra es de tipo máximo y el de venta es de tipo mínimo. Las complejidades para ambos casos son $O(log(n))$ tanto para inserción como para extracción, y para búsqueda del valor máximo o mínimo $O(1)$. Un Heap es una estructura de datos de árbol binario completa donde cada nodo padre tiene un valor menor o mayor que cualquiera de sus hijos, dependiendo del tipo de heap.

En Hsearch se utilizaron árboles AVL, que son árboles binarios de búsqueda auto balanceados donde su mayor propiedad es que la diferencia de alturas entre los subárboles izquiero y derecho de cualquier nodo es como máximo 1. Estos árboles cumplen con que su inserción y búsqueda tienen complejidad $O(log(n))$.

Informe

Respecto a la implementación en memoria, el arreglo de heads se almacena en memoria contigua para permitir un acceso rápido por índice. En los árboles AVL se implementan como nodos enlazados en memoria dinámica, donde cada nodo contiene un ID, punteros a sus dos hijos y altura del nodo; mientras que los heaps se implementan como arreglos en memoria dinámica, siguiendo la estructura de heap binario. La estructura del árbol se mantiene balanceada mediante rotaciones y la propiedad del heap se mantiene mediante percolate_up y percolate_down.

La complejidad de las operaciones relevantes de Heap Market se detalló en la Motivación. Igual podría ser importante mencionar que la ejecución de órdenes tiene complejidad $O(log(n))$ para cada orden, en el peor de los casos.

Respecto a consideraciones y estrategias en los árboles de búsqueda AVL, se supone que estos se mantienen balanceados automáticamente después de cada inserción o eliminación utilizando rotaciones simples y dobles para restaurar el balance. El balance determina qué rotaciones deben realizarse para mantener la estructura balanceada.

Sobre el uso normal de los árboles AVL, estos se utilizan en bases de datos para indexación o en sistemas de archivos para la búsqueda eficiente de archivos. Mientras que los heaps se utilizan en algoritmos de búsqueda o en selección de datos para buscar la mediana o máximo de un conjunto de datos.

Conclusión

El programa Heap Market utiliza heaps para lograr un acceso rápido a la información y un manejo eficiente de las órdenes de compra y venta. Mientras que la parte de A Buscar Cabezas utiliza una combinación de estructuras de datos eficientes (arreglos y árboles AVL) para resolver el problema. Además de este tipo de tareas, también se ahondó en la amplia gama de aplicaciones que tienen estas estructuras de datos en otros campos.

Referencias

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). Introduction to algorithms. MIT press.

https://github.com/IIC2133-PUC/2024-1/tree/main/Clases

https://github.com/IIC2133-PUC/Talleres/tree/master/heaps/priority-queue/queue

https://github.com/IIC2133-PUC/Talleres/tree/master/trees/bst/trees/src/trees