Tecnológico de Monterrey

Campus Querétaro

Programación de estructuras de datos y algoritmos fundamentales | Gpo 601

Act 3.4 - Actividad Integral de BST (Evidencia Competencia)

Maestro
Francisco Javier Navarro Barrón

Presenta:

Karla Alejandra Padilla González A01705331

Alejandra Cabrera Ruiz A01704463

29 de octubre de 2022

Árboles Binarios de Búsqueda

Un árbol es una estructura de datos jerárquica, dinámica y su relación entre los elemento es de uno a muchos, que tiene nodos, nodo es cada uno de los elementos en un árbol y un nodo raíz como el primer elemento agregado al árbol, nodo padre se le llama así al nodo predecesor de un elemento, nodo hijo es el nodo sucesor de un elemento y los hermanos que son nodos que tienen el mismo nodo padre, y nodo hoja aquel que no tiene hijos.

En el caso de un árbol binario de búsqueda es un árbol binario donde los hijos izquierdos son menores y los derechos con mayores que el padre, es decir un árbol co la propiedad de que todos los elementos almacenados en el subárbol izquierdo de cualquier nodo x son menores que el elemento almacenado en x, y todos los elementos almacenados en el subárbol derecho de x son mayores que el elemento almacenado en x.

La implementación de un ABB permite que sea muy simple realizar un procedimiento para una búsqueda, ya que mantiene los datos ordenados aunque no de una forma tradicional sino mediante una estructura binaria. Los recorridos en un árbol ABB, por recorrido típicos existen tres maneras: preorden, inorden y postorden, en el recorrido preorden se visita el nodo raíz del árbol, se recorre el subárbol izquierdo y después el subárbol derecho del nodo raíz.

En el caso del recorrido inorden como primor paso el subárbol izquierdo, se visita la raíz del árbol y se termina por recorrer el subárbol derecho, por último un recorrido postorden se recorre el subárbol izquierdo, después subárbol derecho y por último se visita la raíz del árbol.

Reflexión Alejandra Cabrera

El desarrollo de esta evidencia se me hizo una de las más sencillas de implementar debido a la previa experiencia implementando un Max Heap, aunque usamos un min heap previamente diseñado por el profesor Pedro, el funcionamiento de los métodos son extremadamente similares lo que nos facilitó la implementación en la Actividad 3.4. Por otro lado, los árboles binarios nos facilitan la búsqueda y tienen un ordenamiento similar al de Quicksort. No hemos utilizado inFiles ni outFiles en la ejecución de la actividad porque he notado que usar cin y cout donde iría esos comandos hace que la ejecución del código en replit sea más rápida; al trabajar de forma colaborativa eso se vuelve sumamente importante|.

Es útil ver la implementación de estas estructuras en situaciones reales como la planteada aquí, ya que esta implementación resuelve un problema donde se demuestra que cada estructura puede tener una función diferente y cada una tiene características que la hacen ideal para la solución de la situación planteada. En este caso resultando en la optimización al dar como resultado el menor número de comparaciones para la compañía International Seas.

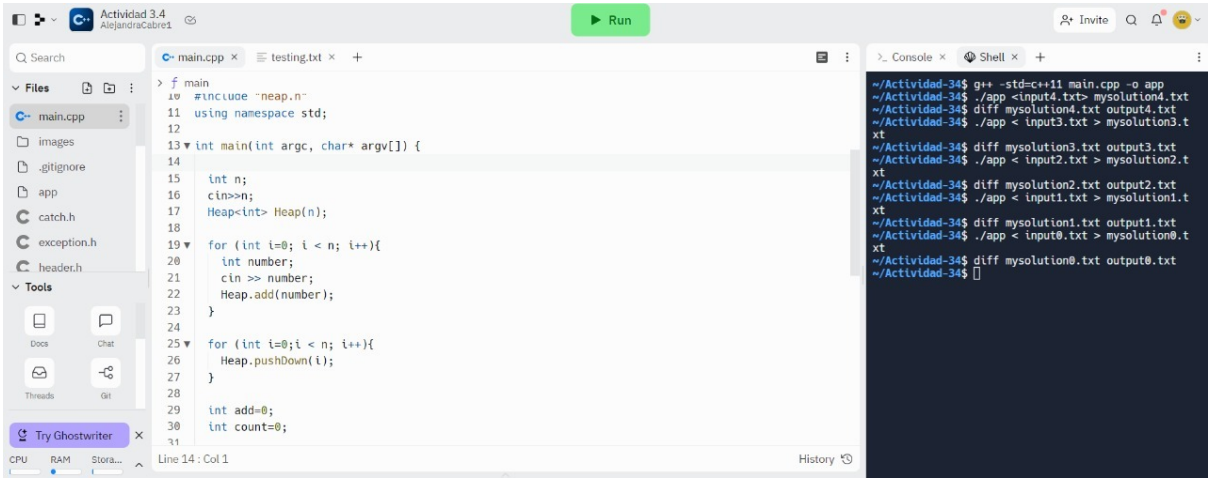
Reflexión Karla Padilla

En la solución de esta actividad con el uso de un árbol binario de búsqueda nos facilitan la búsqueda por distintos tipos de recorridos, y si bien existe uno que es similar al Quicksort que

es inorden donde nos da la lista de nodos de manera ordenada donde el nodo raíz juega un papel similar al del elemento que se tendría en un Quicksort la diferencia que existe enre ellos es que con los ABB existe un mayor gasto de memoria debido a los punteros que son utilizados, sin embargo una de las propiedades que tienen este tipo de árbol es que es fácil diseñar un procedimiento para realizar una búsqueda.

Por último, considero que es muy importante tener en cuenta cualquier tipo de herramienta que puede utilizarse para resolver un problema para que al momento de hacerlo se utilice la más práctica posible para que de esta manera sea más fácil y al final tener un menor espacio de memoria, creo que el poder implementar lo que se ve en clases en situaciones reales nos da la claridad de saber que se puede implementar para resolver un problema cotidiano.

Demostración de los casos de prueba



Referencias

Cursos (s.f) Tema 19: El TAD de las árboles binarios de búsqueda. Recuperado de: https://www.cs.us.es/%7Ejalonso/cursos/i1m/temas/tema-19.html

Universidad de Granada (2022). Árboles binarios de búsqueda. Recuperado de: https://ccia.ugr.es/%7Ejfv/ed1/tedi/cdrom/docs/arb_BB.htm