

Árbol de búsqueda binario autoequilibrado. (AVL tree)

Jose Pablo Herrera Vargas, Miembro del: Tecnológico de Costa Rica,
Esteban Vargas Fernández, Miembro del: Tecnológico de Costa Rica, and
Santiago Chavarria Azofeifa, Miembro del: Tecnológico de Costa Rica.

Abstract—Este documento científico se enfoca en la teoría de los árboles de búsqueda binaria autoequilibrados (AVL). Su lectura es esencial para comprender la diferencia entre los árboles AVL y los árboles de búsqueda binaria estándar (BST), así como para aprender a reconocer y abordar desequilibrios en un árbol AVL debido a inserciones o eliminaciones de nodos. Además, no te pierdas la oportunidad de adentrarte en el mundo de la optimización de datos. Descubre como las fascinantes aplicaciones que conciernen este tipo de estructuras de datos cambian y facilitan nuestra vida cotidiana.

Index Terms—Árboles binarios , AVL, Binary Search Tree

I. INTRODUCCIÓN

LA presente investigación se centra en la teoría referente a los árboles de búsqueda binario autoequilibrados (AVL). Mediante la lectura de los contenidos establecidos se pretende que los estudiantes comprendan el concepto de árboles AVL y los distinguan de los BST. Además, se espera que el estudiante reconozca los tipos de rotaciones que se deben aplicar en caso de algún desequilibrio dentro del árbol a causa de la inserción o eliminación de nodos. [1]

A. Datos históricos

Los árboles AVL llevan su nombre en honor a los matemáticos G. M. Adelson-Velsky y E. M. Landis, quienes inventaron los métodos para construir estos árboles binarios en 1962. Fueron inventados para resolver el problema de los árboles de búsqueda binarios desequilibrados. Entre las operaciones que se realizan con este árbol incluyen: búsqueda, inserción y eliminación [2], [3]

B. Método de funcionamiento

Árbol AVL es una variante especial de un Árbol de Búsqueda Binaria (BST) que se asegura de mantener un equilibrio adecuado en su estructura. Difiere de un BST, puesto que cada nodo del árbol tiene un factor de equilibrio. El factor de equilibrio de un nodo viene dado por el cálculo de la diferencia de alturas entre su subárbol izquierdo y subárbol derecho. Si el factor de equilibrio es -1, 0 o 1 se dice que el nodo está equilibrado, pero con cualquier otro valor se considera que el nodo está desequilibrado. En caso de existir un nodo desequilibrado entonces el árbol también lo estará y requerirá un reequilibrio [4]. Para garantizar este equilibrio, se utilizan operaciones llamadas "rotaciones" cuando se detecta un desequilibrio en el árbol. Estas rotaciones reorganizan los

nodos disminuyendo en el proceso la altura total del árbol en 1 unidad de esta forma el árbol se reequilibra y además, lo hace respetando las propiedades de un BST [5]. Las rotaciones se realizan de cuatro formas diferentes dependiendo del caso en cuestión:

Rotación II: Esta rotación se aplica cuando el factor de equilibrio de un nodo (en este caso C) se vuelve mayor que 1 debido a la inserción de un nodo en el subárbol izquierdo del subárbol izquierdo de C. A esta rotación, también se le conoce como rotación simple a la derecha porque se puede arreglar usando una sola operación de rotación hacia la derecha en el nodo desequilibrado.

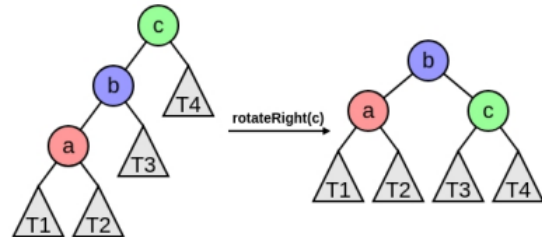


Fig. 1: Rotación simple a la derecha [6].

Rotación DD: Esta rotación es inversa a la rotación LL, puesto que se aplica cuando el factor de equilibrio de un nodo (en este caso A) se vuelve menor que -1 debido a la inserción de un nodo en el subárbol derecho del subárbol derecho de A. También se le conoce como rotación simple a la izquierda porque se puede arreglar usando una sola operación de rotación hacia la izquierda en el nodo desequilibrado.

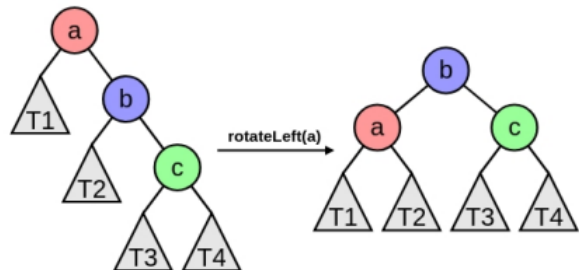


Fig. 2: Rotación simple a la izquierda [6].

Rotación ID: Esta rotación se lleva a cabo cuando el factor de equilibrio de un nodo (en este caso C) se vuelve mayor

que 1 debido a la inserción de un nodo en el subárbol derecho del subárbol izquierdo de C. También se puede ver como una operación compuesta que combina una rotación izquierda y una rotación derecha para restablecer el equilibrio del árbol.

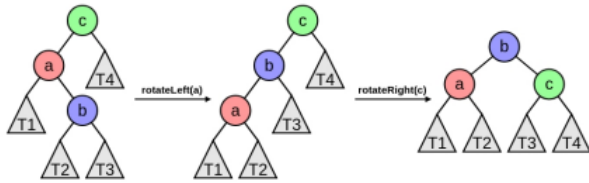


Fig. 3: Rotación izquierda - derecha [6].

Rotación DI: Esta rotación se lleva a cabo cuando el factor de equilibrio de un nodo (en este caso A) se vuelve menor que -1 debido a la inserción de un nodo en el subárbol izquierdo del subárbol derecho de A. La rotación Derecha-Izquierda es una operación compuesta que combina una rotación derecha y una rotación izquierda para restablecer el equilibrio del árbol.

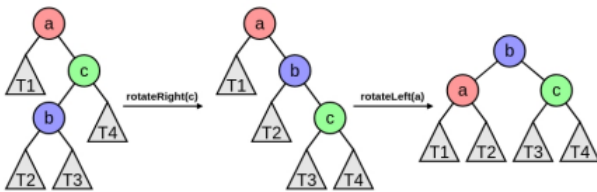


Fig. 4: Rotación derecha - izquierda [6].

C. Método de utilización

Los métodos utilizados en este tipo de árbol, al igual que en el BST, son la inserción y la eliminación. La parte de la codificación en la inserción utiliza la misma lógica que en el inicio: mayores a la derecha y menores a la izquierda. La diferencia radica en que, después de la inserción, se evalúa y compara la altura del árbol desde el nodo recién incluido hasta la raíz principal del árbol, y, en caso necesario, se realizan rotaciones. Lo mismo ocurre en el caso de la eliminación, donde se utilizan los mismos métodos que en un BST convencional, además de la evaluación y comparación de alturas de cada subárbol para llevar a cabo las rotaciones necesarias.

Gracias a esta disposición, funciones como máximo o mínimo requieren menos ciclos para obtener un resultado, de igual manera, según [7], el árbol AVL se utiliza recientemente en muchas aplicaciones; se utiliza en entornos de cómputo en la nube para la programación de tareas. [8] Propone un algoritmo para indexar las palabras clave extraídas de documentos web junto con su contexto, basado en un árbol AVL. Se ha utilizado en redes de sensores inalámbricos para proporcionar un protocolo de seguridad. Además, se ha implementado esta estructura en la clasificación de minería de datos para la inducción de un árbol de decisiones.

II. DEMOSTRACIÓN

Para ejemplificar como se ve y funcionan los arboles AVL, se da la opción de usar esta pagina de visualización de los mismos: [9]

III. CONCLUSIÓN Y RECOMENDACIONES

Se determina que la disposición y eficiencia de los árboles AVL tiene un impacto significativo en diversas aplicaciones de la vida, son una opción poderosa para mantener árboles binarios de búsqueda equilibrados, pero tienen ciertas limitaciones en términos de consumo de recursos y complejidad de implementación. Su elección debe basarse en los requisitos específicos de su aplicación y en las operaciones que realiza con mayor frecuencia.

En caso de necesitar material adicional para una mejor comprensión del tema se recomienda observar el siguiente video que resume los contenidos desarrollados: [10]

REFERENCES

- [1] Catalina Mostaccio, Gabriela Pérez. *Experiencia en la construcción de Objetos de Aprendizaje para árboles AVL usando CROA*, LIFIA – Facultad de Informática de la Universidad Nacional de La Plata, 2015. [Consultado 15-oct-2023]
- [2] "Introduction to AVL trees", *Section*, Disponible en: <https://www.section.io/engineering-education/introduction-to-avl-trees/> [En línea] [Consultado 14-oct-2023]
- [3] Malik Davender S, *Estructuras de datos con C++ (2a. ed.)*, Cengage Learning, 2013. [Consultado 14-oct-2023]
- [4] Kim S. Larsen .*AVL Trees with Relaxed Balance* University of Southern Denmark, 2000. [Consultado 16-oct-2023]
- [5] Tutorialspoint. *Data Structure and Algorithms* [En línea], 2023. [Consultado 16-oct-2023]
- [6] M. Méndez, *Algoritmos y Programación II Tdd Arbol.*, Facultad De Ingeniería, Universidad de Buenos Aires, 2020. [Consultado 16-oct-2023]
- [7] Bounif, Lynda, Djamel Eddine, Zegour. *AVL and Red-black trees as a single balanced tree* 2016. [Consultado 16-oct-2023]
- [8] Tyagi, Nidhi, Rahul Rishi, and R. P. Aggarwal .*Context based Web Indexing for Storage of Relevant Web Pages* International Journal of Computer Applications Vol40, 2012. [Consultado 16-oct-2023]
- [9] Demostración, Disponible en: <https://cs221viz.netlify.app/src/avl>, [En línea]
- [10] Algoritmos Fiuba Curso Buchwald, Disponible en: <https://www.youtube.com/watch?v=PMcQvSM1OAM>, [En línea]