Árboles biselados

Alberto Benavides
18 de marzo de 2020

Los **árboles biselados** (o *splay* en inglés) son un tipo de estructura de datos equivalente a los **árboles binarios**. Éstos son estructuras de datos en los que cada uno de sus elementos tiene la posibilidad de almacenar otros dos elementos (de aquí lo de binarios) denominados **hijo izquierdo** e **hijo derecho**.

En este ámbito, se denomina **raíz** al elemento que no es hijo de ningún otro elemento y **hoja** a los elementos que sólo tienen un hijo o ninguno; a los elementos también se les llama **nodos** o **vértices**; **camino** es el recorrido para ir de un elemento a otro mientras que su **longitud** es la cantidad de nodos que se visitan; el **nivel** de un elemento es igual a la longitud desde el elemento raíz; la **altura** de un árbol es la longitud máxima que hay entre este árbol y sus hojas.

A partir de estas definiciones se pueden enunciar algunas propiedades, tales como:

- El máximo de elementos que puede haber en el nivel k de un árbol binario es 2^k;
- la cantidad máxima de ´vertices en un árbol de altura h es $2^{h+1} - 1$.

Con esto, se puede retornar a la definición de árboles biselados y especificar que son como árboles binarios pero con la peculiaridad de que cada búsqueda convierte al elemento buscado (si se encuentra) en raíz del árbol. Esto se logra aplicando **rotaciones** hacia la izquierda si el elemento buscado tiene un padre de valor inferior, o hacia la derecha en caso contrario. Esta operación se realiza hasta que el elemento buscado llegue a la raíz del árbol. Un ejemplo interactivo escrito en `Javascript` sobre `HTML` se puede manipular en `http://rawcdn.githack.com/jbenavidesv87/algoritmos/master/Actividad%208/index.html`.

Dado este acomodo, la complejidad computacional para buscar el mismo elemento que la útlima búsqueda es $\mathcal{O}(1)$, mientras que en el peor de los casos es $\mathcal{O}(\log(n))$.

Por estos motivos, los árboles biselados son muy utilizados en situaciones donde se esperan búsquedas recurrentes de ciertos elementos, entre las que se pueden mencionar buscadores como Google, enrutadores que deciden entre IPs frecuentes para enviar paquetes, almacenamiento de caché (de hecho, este es uno de sus usos principales por el principio mismo de los cachés) [4].

Referencias

[1] Logan Ingalls. Splay tree implementation in Python. https://gist.github.com/Plutor/3615362, 2012. [Accedido 17/marzo/2020].

[2] Matematicas Discretas. Capítulo 12: TEORIA DE ARBOLES BINARIOS. https://medium.com/@matematicasdiscretaslibro/cap%C3%ADtulo-12-teoria-de-arboles-binarios-f731baf470c0, 2017. [Accedido 17/marzo/2020].

[3] Elisa Schaeffer. Árboles Binarios. https://elisa.dyndns-web.com/teaching/mat/discretas/ejemplos/arboles/arboles.html, 2019. [Accedido 17/marzo/2020].

[4] Shashank Singh. What are some practical applications of AVL trees and splay trees? https://www.quora.com/What-are-some-practical-applications-of-AVL-trees-and-splay-trees, 2016. [Accedido 17/marzo/2020].