

¿Qué son?

Los árboles B+ son una variante de los árboles B, mismos que son una generalización de los árboles balanceados. Los árboles B+ tienen prácticamente la misma estructura que los árboles B con la excepción de que las páginas hoja contienen la clave que las identifica (aquella que se encuentra en la página anterior-superior).

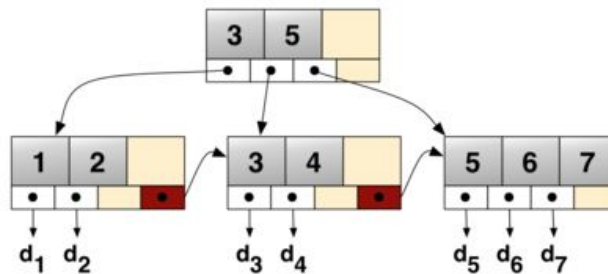


Ilustración 1. Ejemplo de un árbol B+ de orden 2.
Obtenido de: <https://commons.wikimedia.org/wiki/File:Bplustree.png>.

Un árbol B+ de orden d formalmente se define por las siguientes características¹:

1. Cada página, excepto la raíz, contiene m elementos (m es un valor entre d y $2d$).
2. La raíz contiene de 1 a $2d$ elementos.
3. Cada página, excepto la raíz, tiene entre $d + 1$ y $2d + 1$ descendientes.
4. La página raíz tiene al menos dos descendientes.
5. Las páginas hojas están todas al mismo nivel.
6. Toda la información, con las claves que las identifican, se encuentra en las páginas hoja.
7. Las claves almacenadas en las páginas raíz e interiores se utilizan como índices.

¹ "Estructuras de datos", Osvaldo Cairó, Silvia Guardati, 3a ed. México, D. F.: McGraw-Hill Interamericana Editores, c2006.

¿Para qué sirven?

Los árboles B+ se utilizan para la organización de archivos indizados ya que toda la información se encuentra en las hojas (mientras que los nodos raíz e interiores almacenan únicamente las claves).

Los árboles B+ se pueden usar para almacenar información ordenada, la cual es fácil de recuperar recorriendo el árbol.

Los árboles B+ tienen la misma complejidad que los árboles B pero requieren mayor espacio de memoria (ya que en ocasiones almacenan hasta el doble de elementos).

En términos de complejidad, en comparación con otras estructuras de datos, un árbol B+ recorre, busca, inserta y elimina similar a un árbol balanceado (ver tabla 1). Sin embargo, un árbol B+ requiere más espacio en la memoria que un árbol AVL.

Complejidad (Peor Caso)				
	Access	Search	Insertion	Deletion
<u>Array</u>	$O(1)$	$O(n)$	$O(n)$	$O(n)$
<u>Stack</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$
<u>Queue</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$
<u>Singly-Linked List</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$
<u>Doubly-Linked List</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$
<u>Skip List</u>	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>Hash Table</u>	N/A	$O(n)$	$O(n)$	$O(n)$
<u>Binary Search Tree</u>	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>B-Tree [B+ Tree]</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$
<u>AVL Tree</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$

Tabla 1. Comparación de la complejidad de diferentes estructuras de datos.

Ejemplos del uso de árboles B+

Ejemplo 1.

- **El server de SQL usa un árbol B+ para hacer sus búsquedas.**

En el SQL Server, los índices se organizan como árboles B+. Cada página en un índice de árbol B+ es llamada un nodo indexado. Una vez creado el índice, la base de datos lo mantiene automáticamente. Se aplican cada insert, delete y update al índice y se conserva el árbol equilibrado, lo que genera una sobrecarga de mantenimiento para las operaciones de escritura.

Ejemplo 2.

- El sistema operativo de Windows usa arboles B+ para organizar data en el disco duro. NTFS, es el archivo del sistema usado por todos los sistemas operativos de Windows modernos para indexar los directorios.

Ejemplo 3.

- El ReiserFs es un sistema de archivos de propósito general, actualmente soportado por Linux.
- ReiserFS almacena metadatos sobre los ficheros, entradas de directorio y listas de inodos en un único árbol B+ cuya clave principal es un identificador único. Los bloques de disco asignados a los nodos del árbol son los "bloques internos formateados" y los bloques de las hojas son los "bloques de hojas formateados". Todos los bloques restantes son los "bloques sin formatear", que contienen los datos de los ficheros. Los directorios con muchas entradas, ya sean directas o indirectas, que no caben en un sólo nodo, se reparten con el nodo vecino de la derecha.
- En contraste, ext2 y otros sistemas de ficheros, usan una fórmula fija para calcular localizaciones de nodos, por lo que limitan el número de archivos que pueden almacenar. Otros también almacenan los directorios como una simple lista de entradas, lo que provoca que las búsquedas y modificaciones sean operaciones lineales temporalmente y degradan el rendimiento de directorios con muchos archivos. El árbol B+ en ReiserFS evita estos problemas.

Referencias:

- B+ Trees. (n.d.). Retrieved April 13, 2019, from <https://stepik.org/lesson/31576/step/13?unit=11820+>
- [B+ Trees] [Blog post]. (n.d.). Retrieved April 13, 2019, from <http://bigocheatSheet.com>
- Colaboradores de Wikipedia. (2019, January 21). ReiserFS - Wikipedia, la enciclopedia libre. Retrieved April 13, 2019, from <https://es.wikipedia.org/wiki/ReiserFS>