

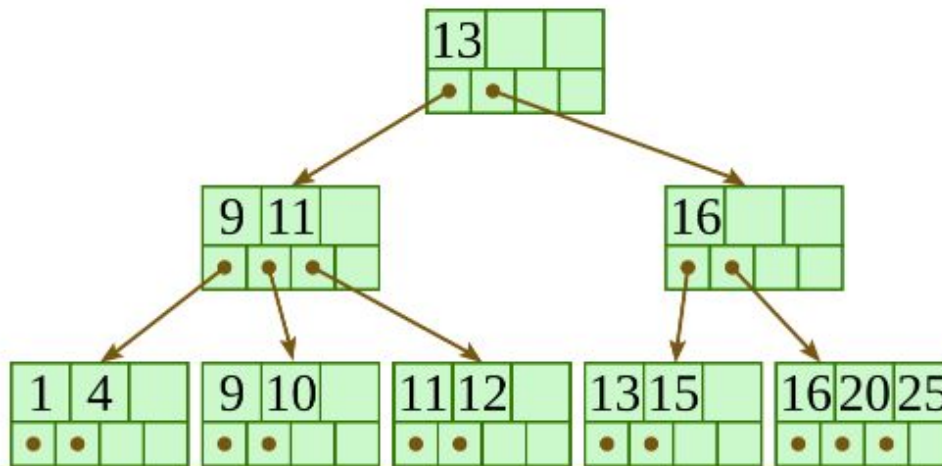
Árboles B+

Los Árboles B+ son una variación de los Árboles B donde los datos están almacenados en las hojas del árbol. Los nodos internos del árbol, es decir, aquellos que no son hojas, consisten de claves de búsqueda y apuntadores que apuntan a niveles inferiores en el árbol, y todas las hojas apuntan a la siguiente hoja en orden de izquierda a derecha. Los Árboles B+ también se balancean para que todas las hojas queden en el mismo nivel y almacenan más de una llave y apuntador en cada nodo. Esto se hace para asegurar que la búsqueda, inserción y borrado tengan complejidad $O(\log n)$. Al buscar un valor dentro de el árbol se compara con la llave dentro de cada nodo, comenzando por la raíz. Si el valor almacenado en el nodo es menor al buscado se sigue el apuntador izquierdo, y si es mayor se sigue el apuntador derecho. Esto asegura que el árbol pueda almacenar cantidades enormes de data sin que la complejidad de la búsqueda crezca exponencialmente, y sin que la altura del árbol en sí crezca demasiado. También cabe notar que los Árboles B+ se pueden utilizar para almacenar datos en memoria RAM, igual que los Árboles B. Todo esto logra que cada nodo de un Árbol B+ pueda apuntar a todos los elementos dentro de un bloque de memoria, y permite realizar búsquedas en sistemas que almacena cantidades enormes de datos de forma rápida y eficiente

Para mantener su balance y función los Árboles B+ deben cumplir con las siguientes características:

- ❑ Todos los nodos deben tener una cantidad de apuntadores iguales a $nk+1$, donde nk es el número máximo de claves en el nodo.
 - ❑ Es decir, si un nodo puede contener hasta 3 claves deberá tener 4 apuntadores
- ❑ Todas las hojas deben estar en el mismo nivel
- ❑ La raíz debe tener un mínimo de dos hijos, excepto cuando solo hay un dato dentro de el árbol
- ❑ Todo nodo tiene un número máximo de hijos iguales a la cantidad de apuntadores posibles en cada nodo, es decir, $nk+1$
- ❑ Todo nodo que no es la raíz ni hoja debe tener un número de hijos mayor o igual a: $(nk+1)/2$
- ❑ Todas las claves deben estar almacenadas dentro de las hojas, en secuencia ordenada de izquierda a derecha
- ❑ Todas las hojas deben tener un número de hijos mayor o igual a $nk/2$ y menor o igual a nk

Fig.1



Para ilustrar la eficiencia de los Árboles B+ podemos utilizar la Figura 1. Los nodos de este Árbol tienen un valor de n_k igual a 3, por ende tienen 4 apuntadores, un mínimo de 2 hijos y un máximo de 4. Usando el Árbol B+ que se muestra en la Figura 1 un usuario puede encontrar los valores con la clave 10 y la clave 15 con el mismo número de comparaciones, pero si se fuera a almacenar los mismos datos en forma de arreglo ordenado se tendría que recorrer la mayoría de la lista para llegar al valor 15.

El uso más común de los Árboles B+ es la administración de metadatos en filesystems y en bases de datos relacionales. Ejemplos notables de los filesystems que usan Árboles B+ incluyen a ReiserFS, NSS, XFS, JFS, ReFS y BFS donde se usan para almacenar metadatos, adicionalmente Microsoft SQL Server, IBM DB2, Informix y Oracle 8 tienen la opción de organizar índices de tablas en forma de Árbol B+.

Referencias:

ali Dana, H. (2017). B+ tree intro, uses, insertion and deletion. [en línea] Slideshare.net. Disponible en: <https://www.slideshare.net/HAMID-50/b-tree-introusesinsertion-and-deletion> [Recuperado 5 Apr. 2018].

Cburch.com. (n.d.). CSci 340: B+-trees. [en línea] Disponible en: <http://www.cburch.com/cs/340/reading/btree/index.html> [Recuperado 5 Apr. 2018].

Cs.usfca.edu. (n.d.). B+ Tree Visualization. [en línea] Disponible en: <https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html> [Recuperado 8 Apr. 2018].

en.wikipedia.org. (n.d.). B+ tree. [en línea] Disponible en: https://en.wikipedia.org/wiki/B%2B_tree [Recuperado 6 Apr. 2018].

Harris, M. (2010). What is a B+ Tree?. [en línea] Quora.com. Disponible en: <https://www.quora.com/What-is-a-B+-Tree> [Recuperado 5 Apr. 2018].