

AED Tema 5. Árboles equilibrados B y B+

Martín González Dios 

29 de enero de 2025

Si queremos almacenar una cantidad muy grande de datos, podemos llegar al caso de que ocupe varios bloques de disco, lo que supone tener que hacer varios accesos, ralentizando mucho el tiempo de búsqueda. Para minimizar el problema del tiempo de acceso al disco (u otros medios externos), se desarrollaron los **árboles B** y **árboles B+**.

1. Árboles B

Arboles siempre equilibrados que minimizan el tiempo de búsqueda y borrado de elementos. En ellos, los nodos se agrupan en **páginas** (unidad a la que se accede en bloque, almacena la info. de un grupo de nodos identificados por claves). En un árbol B de **orden m** todas las páginas internas tienen entre $\lceil m/2 \rceil$ y m ramas no vacías, y la raíz tiene 0 (si es la única página) o entre 2 y m ramas. En cada página interna hay (ramas - 1) claves, las cuales dividen las claves de sus ramas como un árbol de búsqueda. Todas las páginas hoja se encuentran en el mismo nivel, por eso es un **árbol totalmente equilibrado**.

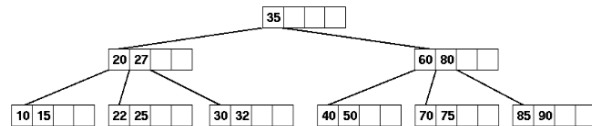


Figura 1: Representación de árbol B

1.1. Inserción

Se busca la página correspondiente, descendiendo por el árbol hasta llegar a la página hoja adecuada. Al querer insertar en dicha hoja, se pueden dar 2 casos:

- Si la página no está llena, se inserta el elemento.
- Si la página está llena, ésta se divide en 2 (insertando ya el elemento en su sitio correspondiente) y el elemento mediano (el que ocupe la posición del medio) asciende en el árbol al nivel superior, insertándose en la página padre, y de dicho elemento saldrán dos ramas, una a cada página generada en el elemento inferior (eliminándose, evidentemente, la rama que apuntaba a la página que se acaba de dividir, y que por tanto dejó de existir).

1.2. Búsqueda

Para buscar una clave en el árbol B se necesita partir de la página raíz y, evidentemente, especificar la clave a buscar. Por tanto, estos serán los dos parámetros que se le pasen a la función `Buscar()`. Se necesita un procedimiento auxiliar llamado `Buscarnodo()` que examina para cada página dada si la clave se encuentra en ella o, en caso contrario, entre que dos claves iría. En el segundo de los casos, se desciende a la página correspondiente del nivel inferior por la rama que se encuentra entre los elementos, y se repite la búsqueda.

1.3. Eliminación

En las eliminaciones también se pueden dar varios casos:

- **La página donde se va a eliminar tiene más elementos que el mínimo:** si es una página hoja, se elimina sin problema. Si no es hoja, se sustituye el elemento por el predecesor en la hoja que salga de su rama derecha, y se elimina el elemento ahora en la página hoja.
- **La página donde se va a eliminar tiene el número mínimo de elementos:** se pueden dar 2 casos:
 - Si una de las hojas hermanas tiene más elementos que el mínimo, se toma el elemento del extremo de dicha página, se sube a la padre, y de la padre se baja el correspondiente a la página en la que queremos hacer la eliminación, para tener más elementos que el mínimo y poder hacer la eliminación simple.
 - Si todas las páginas hermanas tienen el mínimo de elementos, se unifican las dos páginas y se trae la mediana de ambas que está en la página padre. De esta forma la página padre reduce en uno sus elementos, y también en una de sus ramas.

2. Árboles B*

Optimización de los árboles B. En la inserción, si la página está llena, se mueven elementos a una página hermana, posponiendo la división hasta que los hermanos estén completos. Con esto se reducen el número de divisiones.

3. Árboles B+

Optimización de los árboles B para conseguir un recorrido secuencial más rápido. Todos los elementos se encuentran en páginas hoja, de forma que algunos de estos elementos aparecen duplicados en las páginas de niveles superiores únicamente con la función de actuar como índices. Debido a esto, ocupan algo más de espacio en memoria, pero para árboles que se modifican frecuentemente supone también un aspecto positivo, ya que evita la reorganización del árbol, que era la operación más costosa en los árboles B.

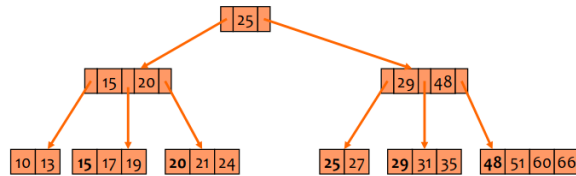


Figura 2: Representación de árbol B+

3.1. Inserción

Por lo general es similar a la inserción en los árboles B, salvo por el caso en el que la página esté llena. En estos casos, el elemento mediana sube a la página padre. Pero también se mantiene en la página de la derecha resultante de la división en 2 de la página hoja original.

3.2. Eliminación

La eliminación es mucho más simple que en árboles B, ya que los elementos siempre están en las páginas hojas. Se pueden dar dos casos:

- Si tras eliminar el elemento la página todavía tiene el mínimo de elementos o más, no hay que hacer ninguna modificación.
- Si tras eliminar el elemento la página queda con menos elementos que el mínimo, hay redistribuir los elementos, borrando además un índice en la página de nivel superior, o cambiándolo.