

## Altas y Bajas en Árboles B y B+

### Propiedades de un Nodo en Árboles B de Orden M

- **Mínimo:** Cada nodo (excepto la raíz) tiene al menos  $\lceil M/2 \rceil - 1$  elementos.
- **Máximo:** Cada nodo puede contener como máximo M descendientes directos (hijos) y M-1 elementos.

### Árboles B

#### Altas:

- **Overflow:** Cuando un nodo alcanza la capacidad máxima, se crea un nuevo nodo. Las claves se distribuyen entre ambos nodos, La segunda mitad de las claves se traslada al nuevo nodo y la menor clave de la segunda mitad se promociona al nodo padre.

#### Bajas:

1. Si la clave a eliminar no está en una hoja, se reemplaza con la menor clave del subárbol derecho.
2. Si el nodo hoja tiene al menos el mínimo número de claves después de la eliminación, no se necesita más acción.
3. En caso contrario, se produce un underflow.

#### Políticas para resolver underflow:

- Se intenta redistribuir con un hermano adyacente. Si no es posible, se fusiona con él. \* La redistribución es un proceso mediante el cual se trata de dejar cada nodo lo más equitativamente cargado posible.
- **Política Izquierda:** Se intenta con el hermano adyacente izquierdo primero. Si no es posible, se fusiona con él.
- **Política Derecha:** Se intenta con el hermano adyacente derecho primero. Si no es posible, se fusiona con él.
- **Política Izquierda o Derecha:** Se intenta con el hermano izquierdo. Si no es posible, se intenta con el derecho. Si ambos fallan, se fusiona con el izquierdo.
- **Política Derecha o Izquierda:** Se intenta con el hermano derecho. Si no es posible, se intenta con el izquierdo. Si ambos fallan, se fusiona con el derecho.
- **Casos Especiales:** Si es un nodo hoja en un extremo, se intenta con su único hermano disponible.

---

### Árboles B+

#### Altas:

- **Dificultad:** Inserción en un nodo lleno (overflow). El nodo se divide en dos, distribuyendo equitativamente las claves. La clave del medio o la menor de las

mayores se promociona al padre. Solo se copia la clave en caso de overflow en una hoja.

### **Bajas:**

- Si al eliminar una clave, la cantidad de claves que queda es mayor o igual que  $\lceil M/2 \rceil - 1$ , se termina la operación.

### **Políticas de Resolución de Underflow:**

- Mismas políticas que en Árboles B.

Ambos tipos de árboles mantienen la estructura balanceada y optimizan la búsqueda y recorrido, con diferencias en la manera de manejar las altas y bajas.

### **Diferencias Clave**

- **Estructura de las Hojas:**
  - En un árbol B+, todas las claves están en las hojas, y los nodos internos solo actúan como índices.
  - En un árbol B, las claves pueden estar tanto en los nodos internos como en las hojas.
- **Promoción y Copia de Claves:**
  - En los árboles B+, al dividir un nodo hoja, se copia una clave al nodo padre, manteniendo la clave original en el nodo hoja.
  - En los árboles B, la clave promocionada se mueve al nodo padre y no se mantiene una copia en el nodo hoja.

### **Resumen**

- **Similitud:** En ambos tipos de árboles, la inserción puede causar un overflow que resulta en la división del nodo afectado y la promoción de una clave al nodo padre.
- **Diferencia:** En árboles B+, las claves se copian en los nodos internos durante una inserción en una hoja, mientras que en árboles B, la clave promocionada se mueve al nodo padre sin mantener una copia en la hoja.

Estas diferencias reflejan las optimizaciones en los árboles B+ para soportar recorridos secuenciales más rápidos, mientras que los árboles B están diseñados para mantener un equilibrio general en la estructura del árbol.