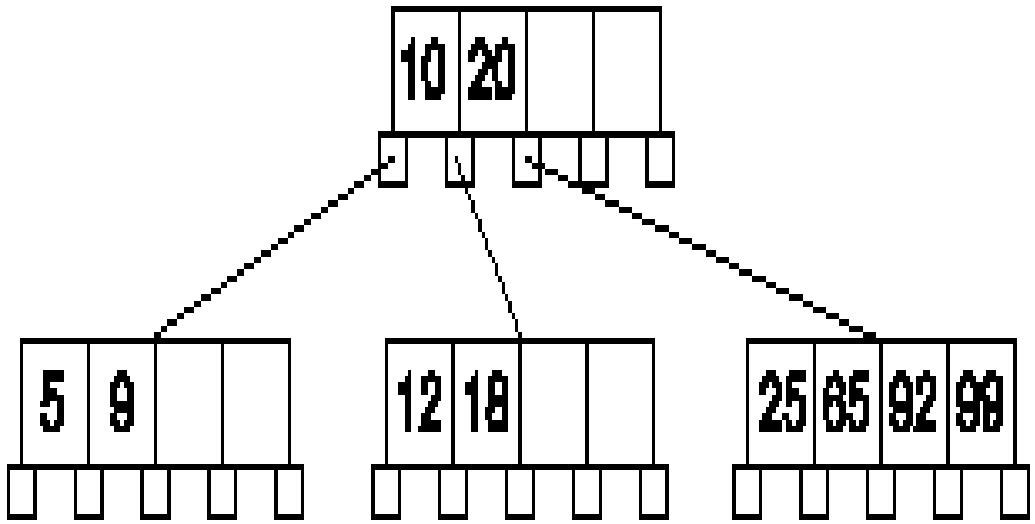
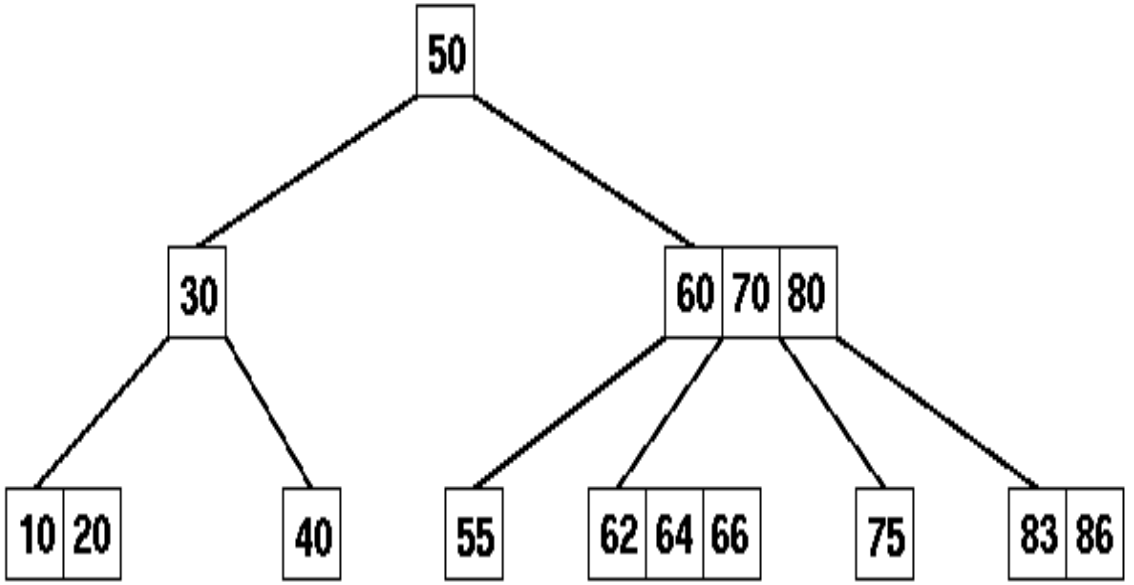
# ARBOLES B

# ¿QUÉ SON?

Un árbol-B tiene la capacidad de almacenar mas de un elemento en un nodo, debe estar totalmente balanceado, a comparación de un árbol AVL este puede tener mas de hijos, se mantiene balanceado porque requiere que todos los nodos hoja se encuentren a la misma altura y es considerado como un árbol multicamino.




# EJEMPLO DE UN ÁRBOL B

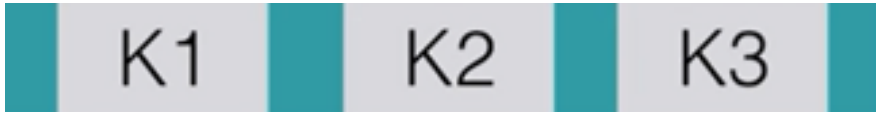
# ¿PARA QUE SIRVE?

El árbol B es usado para guardar grandes cantidades de información debido a su capacidad de poder almacenar mas de un elemento en un nodo.

# ¿CÓMO SE CONSTRUYE?

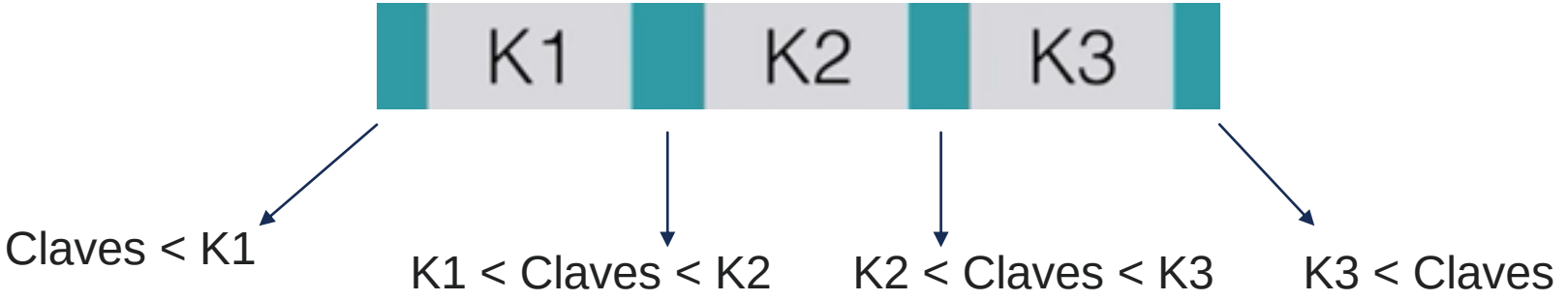
Un árbol B se crea con nodos que tienen los componentes:

- Claves
- Apuntadores a hijos




3 claves (K1, K2, k3)
4 Hijos

Las claves del nodo siempre ordenadas de menor a mayor; K1 < K2 < K3.
Y los hijos del nodo siempre tendrán todas sus claves mayores o todas menores respecto a una de las claves del nodo padre, dependiendo del apuntador a hijo del que se trate, es decir:



# ¿CÓMO SE CONSTRUYE?

un árbol-B cumple con las siguientes características:

> Su orden es el número máximo de ramas que pueden partir de un nodo.
> Si de un nodo n ramas, ese nodo contendrá n-1 claves.
> El árbol está ordenado.
> Todos los nodos terminales, (nodos hoja), están en el mismo nivel.
> Todos los nodos intermedios, excepto el raiz, deben tener entre m/2 y m ramas no nulas.
> El máximo número de claves por nodo es m-1.
> El mínimo número de claves por nodo es (m/2)-1.
> La profundidad (h) es el número máximo de consultas para encontrar una clave

la forma más eficiente de construir el árbol-B inicial es construir el conjunto inicial de nodos hoja directamente desde la entrada, y después construir los nodos internos a partir de este conjunto. Inicialmente, todas las hojas excepto la última tienen un elemento más, el cual será utilizado para construir los nodos internos.

# OPERACIONES DE UN ÁRBOL B

- Búsqueda
- Insertar elemento
- Eliminar elemento
- rebalanceo

## Búsqueda:

La búsqueda es similar a la de los árboles binarios. Se empieza en la raíz, y se recorre el árbol hacia abajo, escogiendo el sub-nodo de acuerdo a la posición relativa del valor buscado respecto a los valores de cada nodo. Típicamente se utiliza la búsqueda binaria para determinar esta posición relativa.

# OPERACIONES DE UN ÁRBOL B

## Insertar elemento:

Las inserciones se hacen en los nodos hoja, Si las divisiones de nodos suben hasta la raíz, se crea una nueva raíz con un único elemento como valor separador, y dos hijos. Es por esto por lo que la cota inferior del tamaño de los nodos no se aplica a la raíz. El máximo número de elementos por nodo es $U$-1. Así que debe ser posible dividir el número máximo de elementos $U$-1 en dos nodos legales. Si este número fuera impar, entonces $U=2L$, y cada uno de los nuevos nodos tendrían $(U-2)/2 = L-1$ elementos, y por lo tanto serían nodos legales. Si $U$-1 fuera par, $U=2L$-1, así que habría $2L$-2 elementos en el nodo.

# OPERACIONES DE UN ÁRBOL B

## Eliminar elemento:

Hay dos formas de eliminar un nodo de un árbol B:

- localizar y eliminar el elemento, y luego corregir, o
- hacer una única pasada de arriba abajo por el árbol, pero cada vez que se visita un nodo, reestructurar el árbol para que cuando se encuentre el elemento a ser borrado, pueda eliminarse sin necesidad de continuar reestructurando.

Se pueden dar dos problemas al eliminar elementos. Primero, el elemento puede ser un separador de un nodo interno. Segundo, puede suceder que al borrar el elemento número de elementos del nodo quede debajo de la cota mínima. Estos problemas se tratan a continuación en orden.

# Dividir:

Ocurre cuando queremos insertar elementos a un nodo que ya está lleno, por ejemplo cuando intentamos insertar elementos en un árbol cuyos nodos ya tienen todos un numero de hijos que concuerdan con el grado del árbol.
En este caso deberemos reequilibrar o rebalancear el árbol de modo que podamos seguir insertando datos… Básicamente consiste en reorganizar el árbol para que se encuentren huecos de alguna manera

# Rebalanceo:

Si al eliminar un elemento de un nodo hoja el nodo se ha quedado con menos elementos que el mínimo permitido, algunos elementos se deben redistribuir. En algunos casos el cambio lleva la deficiencia al nodo padre, y la redistribución se debe aplicar iterativamente hacia arriba del árbol, quizá incluso hasta a la raíz. Dado que la cota mínima en el número de elementos no se aplica a la raíz, el problema desaparece cuando llega a esta.

# GRACIAS

EQUIPO 5