

Algoritmos y estructuras de datos

Árboles balanceados

Francisco Javier Zaragoza Martínez

Universidad Autónoma Metropolitana Unidad Azcapotzalco
Departamento de Sistemas



24 de mayo de 2021

Eurípides

Lo mejor y más seguro es mantener el **balance** en tu vida.

Hazrat Inayat Khan

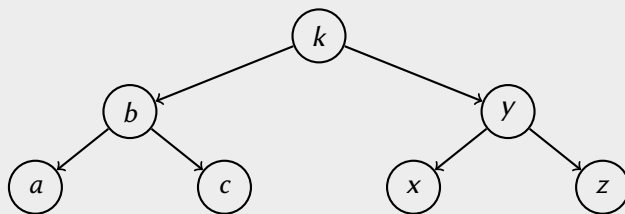
Debe haber un **balance** en todas tus acciones, ser extremo o tibio es igualmente malo.

Elizabeth Gilbert

Para encontrar el **balance** que desea, esto es en lo que debe convertirse. Debe tener los pies tan firmemente en la tierra que sea como si tuviera cuatro piernas en lugar de dos.

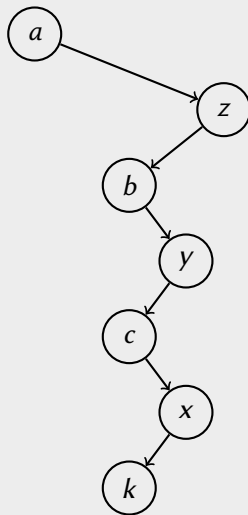
Problema de los árboles binarios de búsqueda

Si nos va bien: altura logarítmica



Problema de los árboles binarios de búsqueda

Si nos va mal: altura lineal



Árboles balanceados

Diremos que un árbol enraizado de orden n es α -balanceado si todas sus hojas tienen profundidad $\leq 1 + \alpha \log_2 n$. Idealmente $\alpha = 1$.

¿Cómo se logra esto?

Una solución al problema de crear árboles desbalanceados es la de reorganizar sus nodos conforme se hagan inserciones y borrados. Algunos ejemplos son:

Árboles AVL árboles binarios de búsqueda 1.44-balanceados.

Árboles rojinegros árboles binarios de búsqueda 2-balanceados.

Árboles 2-3-4 árboles cuaternarios de búsqueda 1-balanceados.

Otros ejemplos incluyen a los árboles B, los árboles *biselados* y los *treaps*.

Árboles 2-3-4

Un **árbol 2-3-4** es un árbol balanceado en el que todas sus hojas tienen la **misma** profundidad y donde sus nodos pueden:

- 1** contener **una** clave x_1 y **dos** árboles A_0 y A_1 .
- 2** contener **dos** claves $x_1 < x_2$ y **tres** árboles A_0, A_1 y A_2 .
- 3** contener **tres** claves $x_1 < x_2 < x_3$ y **cuatro** árboles A_0, A_1, A_2 y A_3 .

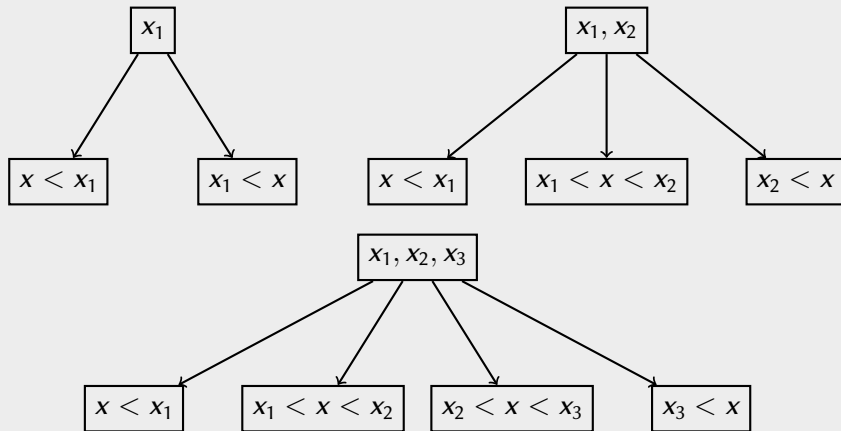
En cada caso se cumple que si a es una clave en el árbol A_{i-1} y c es una clave en el árbol A_i entonces $a < x_i$ y $x_i < c$.

Tipos de nodos

Los nodos con dos, tres y cuatro árboles se llaman 2 nodos (binarios), 3 nodos (ternarios) y 4 nodos (cuaternarios), respectivamente.

Árboles 2-3-4

Nodos binarios, ternarios y cuaternarios



Árboles 2-3-4

Tipos asociados a un árbol 2-3-4

Definiremos un tipo estructurado `nodo234` para representar un nodo y un tipo `arbol234` para representar un árbol 2-3-4. El tipo `nodo234` consiste de un contador, tres datos y cuatro apuntdores a sus sucesores (valdrán `NULL` si son vacíos).

```
typedef struct nodo234 {  
    int t;           // contador de claves  
    int a[3];        // datos almacenados  
    struct nodo234 *p[4]; // enlaces a sucesores  
} nodo234;
```

Por otro lado, el tipo `arbol234` es un apuntador a `nodo234`.

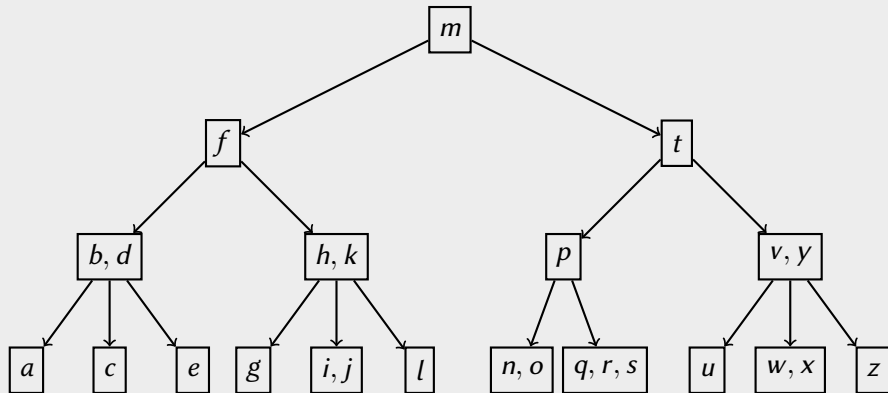
```
typedef struct nodo234 *arbol234;
```

Note que los tipos `nodo234 *`, `struct nodo234 *` y `arbol234` son equivalentes.

Árboles 2-3-4

Ejemplo

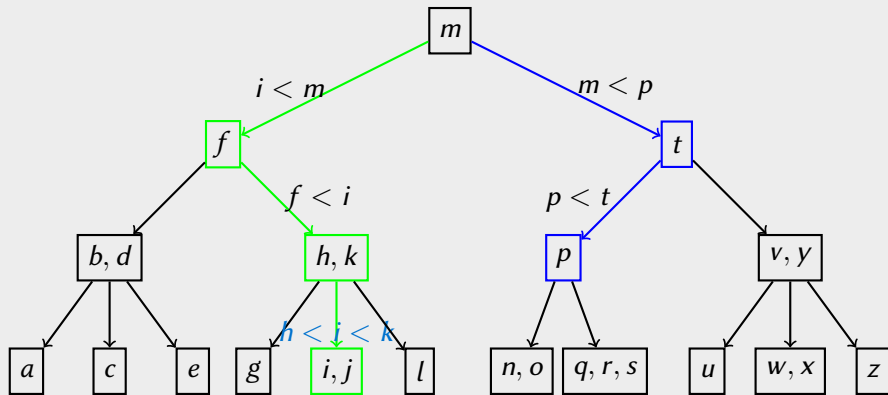
Un árbol 2-3-4 con 11 nodos binarios, 6 ternarios, 1 cuaternario y 11 hojas.



Árboles 2-3-4

Ejemplo de búsqueda

Buscar la i y buscar la p .



Árboles 2-3-4

Inserción en un árbol 2-3-4

Inserción de una clave

- 1 Si la raíz está vacía, se agrega la clave a la raíz y se termina.
- 2 Se hace la búsqueda de la clave con dos salvedades:
 - 1 Si durante la búsqueda se ve un nodo cuaternario, se **reorganiza** y se continúa.
 - 2 Si la clave se encuentra, se termina.
- 3 La clave se agrega a la hoja que se haya llegado. **¿Por qué cabe?**

Reorganizar un nodo cuaternario

- 1 Se manda la clave x_2 al nodo precursor (si era la raíz, se **crea** una nueva raíz). ¿Por qué cabe x_2 en el precursor?
- 2 Se sustituye el nodo cuaternario por dos nodos binarios, con claves x_1 y x_3 y con árboles A_0, A_1 y A_2, A_3 respectivamente.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

q

Las claves q, w entran a la raíz.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

q, w

Las claves q, w entran a la raíz.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

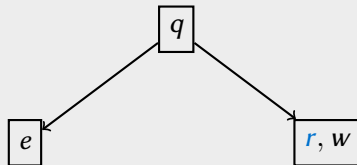
e, q, w

La clave e entra a la raíz, que se vuelve **cuaternaria**.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

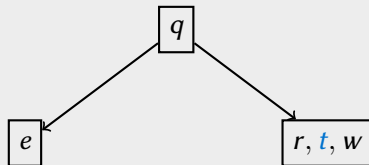


La clave r obliga a reorganizar la raíz.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

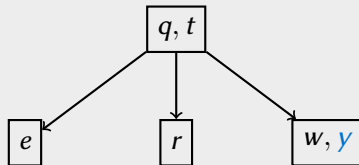


La clave t crea una hoja cuaternaria.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

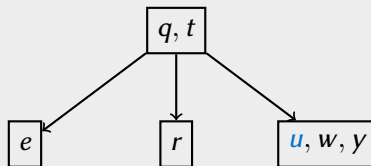


La clave y obliga a reorganizar esa hoja.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

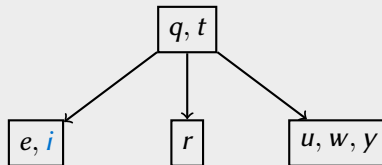


La clave u crea una hoja cuaternaria.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

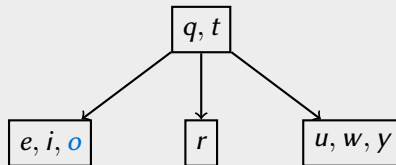


Las claves i, o entran en hojas.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

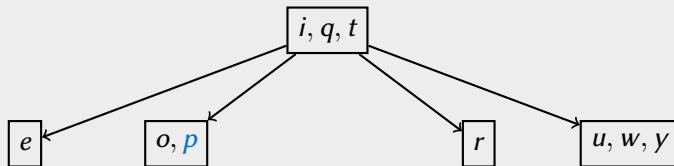


Las claves i, o entran en hojas.

Árboles 2-3-4

Ejemplo de inserción

Queremos insertar claves en el orden $q, w, e, r, t, y, u, i, o, p$.

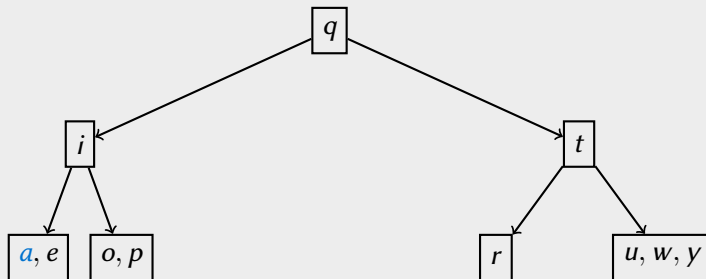


La clave p obliga a reorganizar una hoja. La raíz es cuaternaria.

Árboles 2-3-4

Ejemplo de inserción

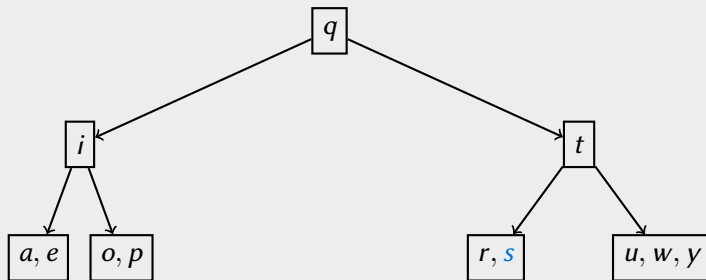
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave a obliga a reorganizar la raíz.



Árboles 2-3-4

Ejemplo de inserción

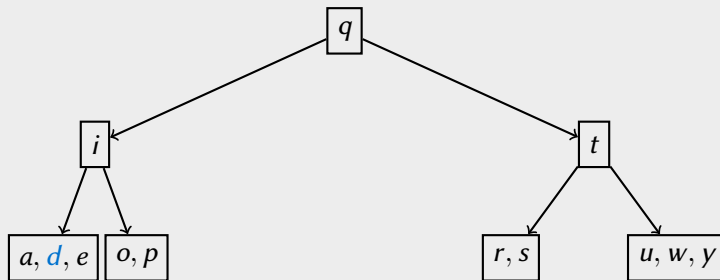
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave s entra en una hoja.



Árboles 2-3-4

Ejemplo de inserción

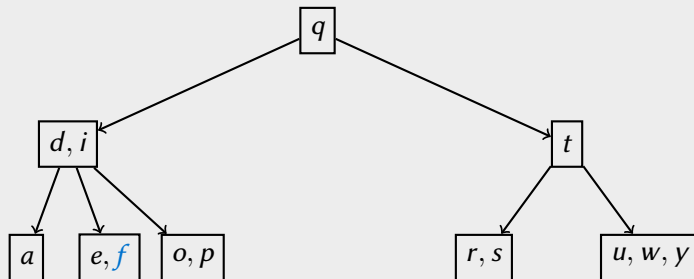
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave d crea una hoja cuaternaria.



Árboles 2-3-4

Ejemplo de inserción

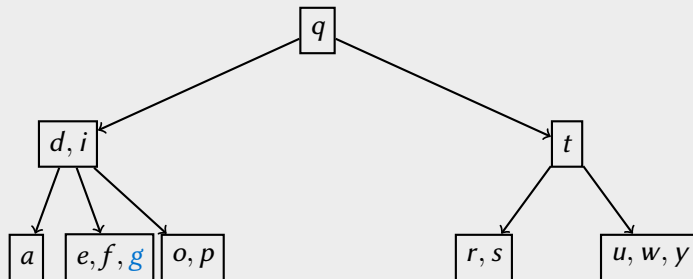
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave f obliga a reorganizar esa hoja.



Árboles 2-3-4

Ejemplo de inserción

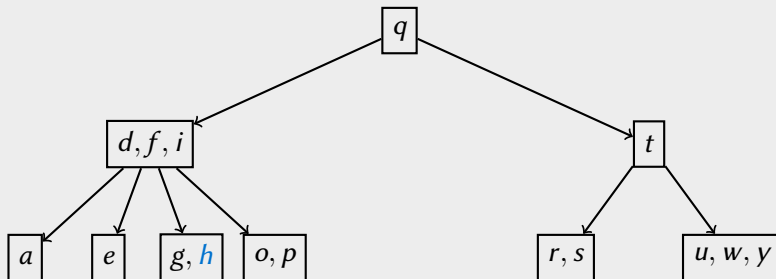
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave g crea una hoja cuaternaria.



Árboles 2-3-4

Ejemplo de inserción

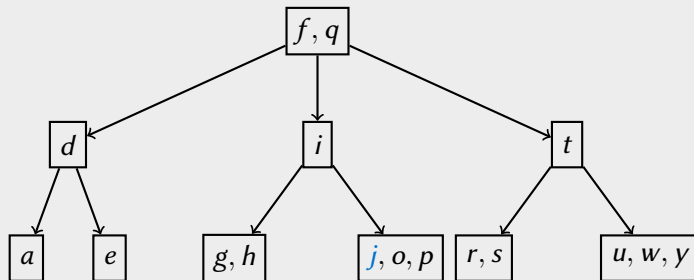
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave h obliga a reorganizar esa hoja y crea un nodo cuaternario.



Árboles 2-3-4

Ejemplo de inserción

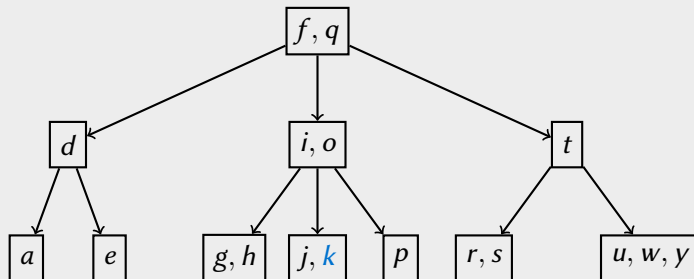
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave j obliga a reorganizar ese nodo y crea una hoja cuaternaria.



Árboles 2-3-4

Ejemplo de inserción

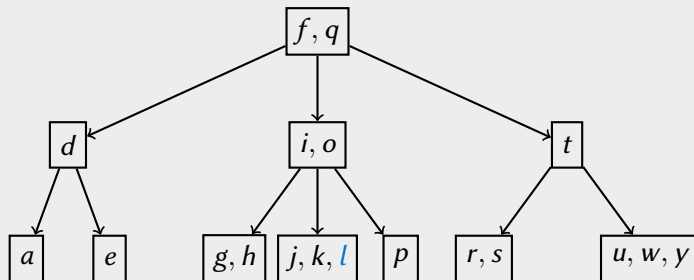
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave k obliga a reorganizar esa hoja.



Árboles 2-3-4

Ejemplo de inserción

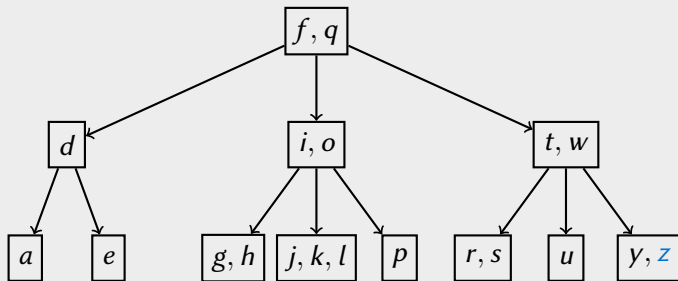
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave l crea una hoja cuaternaria.



Árboles 2-3-4

Ejemplo de inserción

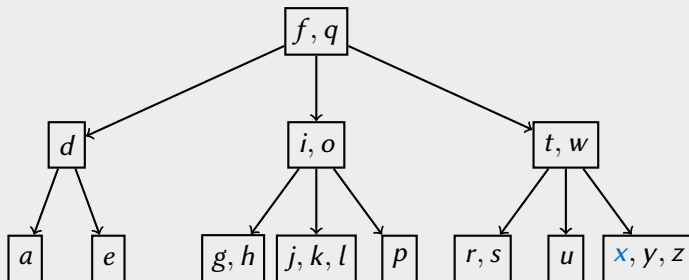
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave z obliga a reorganizar una hoja cuaternaria.



Árboles 2-3-4

Ejemplo de inserción

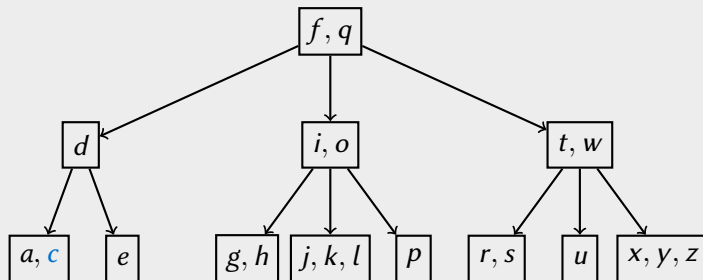
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
Las claves x, c, v entran en hojas.



Árboles 2-3-4

Ejemplo de inserción

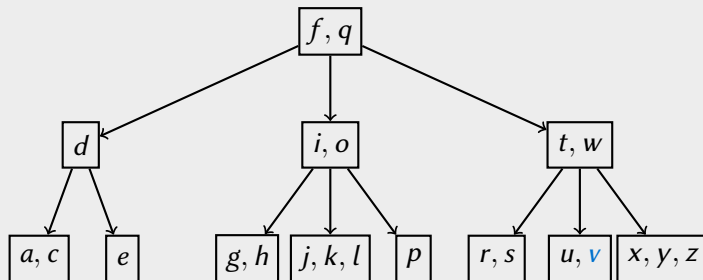
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
Las claves x, c, v entran en hojas.



Árboles 2-3-4

Ejemplo de inserción

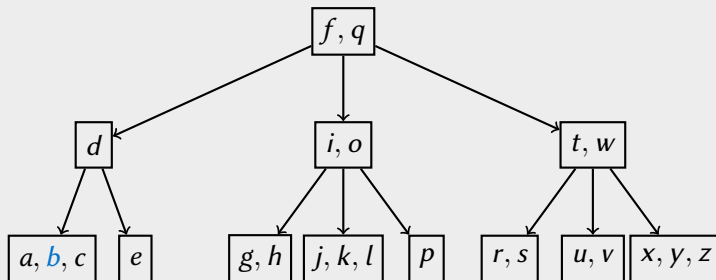
Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
Las claves x, c, v entran en hojas.



Árboles 2-3-4

Ejemplo de inserción

Ahora queremos insertar claves en el orden $a, s, d, f, g, h, j, k, l, z, x, c, v, b$.
La clave b crea una hoja cuaternaria.



Árboles 2-3-4

Propiedades de balance

Balance perfecto

Los árboles balanceados que tienen todas sus hojas a la misma profundidad se dicen **perfectamente** balanceados. Los árboles 2-3-4 son 1-balanceados en el **peor** caso (si todos los nodos fueran binarios) y 0.5-balanceados en el **mejor** caso (si todos los nodos fueran cuaternarios).

Balance de árboles 2-3-4

Ocurre automáticamente: La profundidad de todas las hojas aumenta simultáneamente cuando se reorganiza la raíz (y nunca más).

¿Por qué no se usan?

Las implementaciones son lentas debido a los diferentes tipos de nodos.

Árboles rojinegros

Color de los nodos

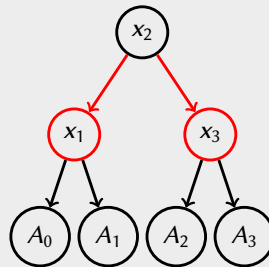
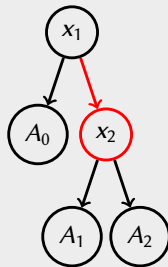
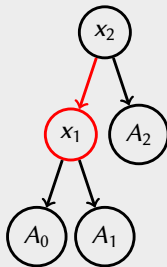
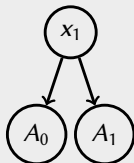
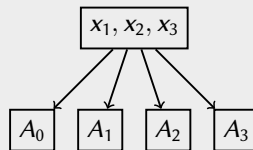
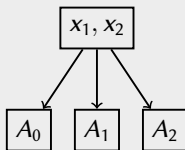
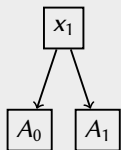
Los árboles 2-3-4 se pueden representar como árboles binarios de búsqueda con un bit adicional por nodo, llamado **color**, que puede ser **rojo** o **negro**. En un árbol rojinegro:

- 1 La raíz siempre es negra.
- 2 Los nodos negros pueden tener sucesores negros o rojos.
- 3 Los nodos rojos sólo pueden tener sucesores negros.

Además, como queremos que sean representaciones de árboles 2-3-4, todos los caminos de la raíz a las hojas tendrán la misma cantidad de nodos negros. La altura de un árbol rojinegro será entonces $\leq 1 + 2 \log_2 n$.

Árboles rojinegros

Representación de los nodos de un árbol 2-3-4



Árboles rojinegros

Tipos asociados a un árbol rojinegro

Definiremos un tipo estructurado `nodoRN` para representar un nodo y un tipo `arbolRN` para representar un árbol rojinegro. El tipo `nodoRN` consiste de un color, un dato y dos apuntadores a sus sucesores (valdrán `NULL` si son vacíos).

```
typedef struct nodoRN {  
    int c;                // color del nodo  
    int a;                // dato almacenado  
    struct nodoRN *izq, *der; // enlaces a sucesores  
} nodoRN;
```

Por otro lado, el tipo `arbolRN` es un apuntador a `nodoRN`.

```
typedef struct nodoRN *arbolRN;
```

Note que los tipos `nodoRN *`, `struct nodoRN *` y `arbolRN` son equivalentes.

Árboles rojinegros

Operaciones en árboles rojinegros

Cada una de las operaciones en los nodos cuaternarios de un árbol 2-3-4 se puede traducir a operaciones en el árbol rojinegro correspondiente.

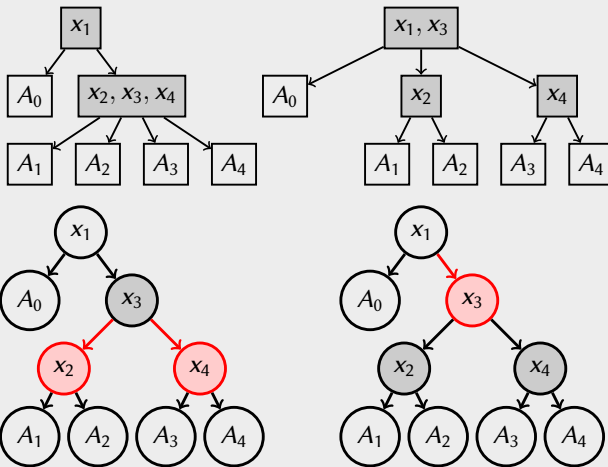
Tipos de operaciones

Estas operaciones se llaman **cambios de color** y **rotaciones**.

- ▶ Los cambios de color ocurren en los casos sencillos. Hay dos tipos:
 - ▶ Precursor binario.
 - ▶ Precursor ternario.
- ▶ Las rotaciones ocurren en los casos complicados. Hay dos tipos:
 - ▶ Rotaciones **simples** izquierda/izquierda y derecha/derecha.
 - ▶ Rotaciones **dobles** izquierda/derecha y derecha/izquierda.
- ▶ Las de cada tipo son simétricas, sólo veremos una de cada tipo.

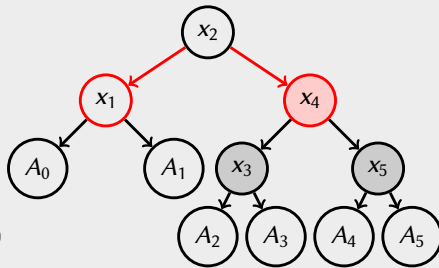
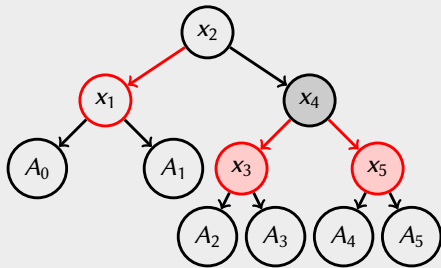
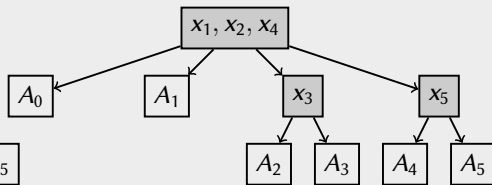
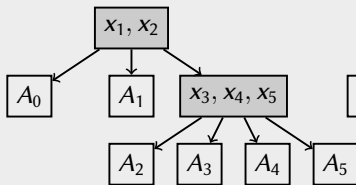
Árboles rojinegros

Nodo cuaternario con precursor binario (cambio de color)



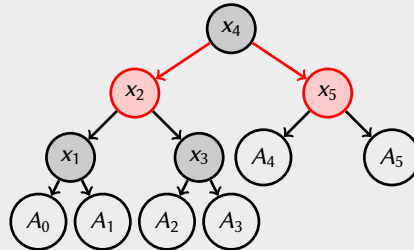
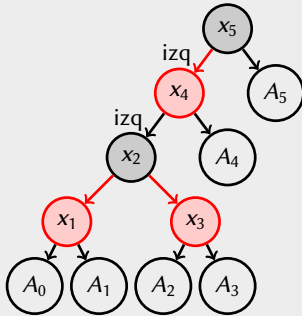
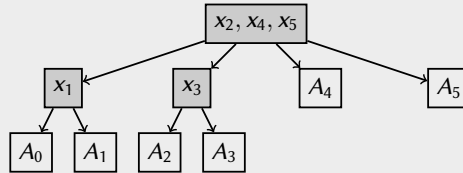
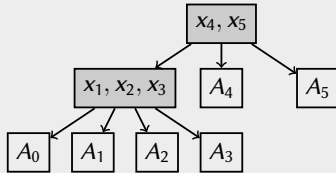
Árboles rojinegros

Nodo cuaternario con precursor ternario (cambio de color)



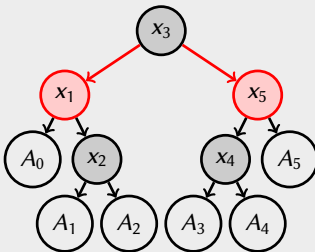
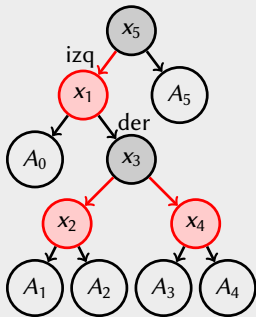
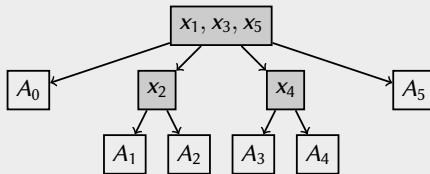
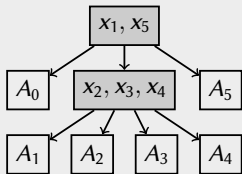
Árboles rojinegros

Nodo cuaternario con precursor ternario (rotación simple izq/izq)



Árboles rojinegros

Nodo cuaternario con precursor ternario (rotación doble izq/der)



Árboles rojinegros

Ejemplo de inserción

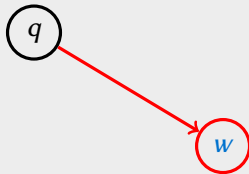
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Raíz.



Árboles rojinegros

Ejemplo de inserción

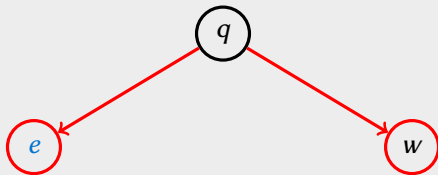
Claves en orden q , w , e , r , t , y , u , i , o , p , a , s . Hoja.



Árboles rojinegros

Ejemplo de inserción

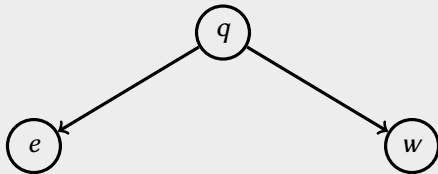
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja y raíz llena.



Árboles rojinegros

Ejemplo de inserción

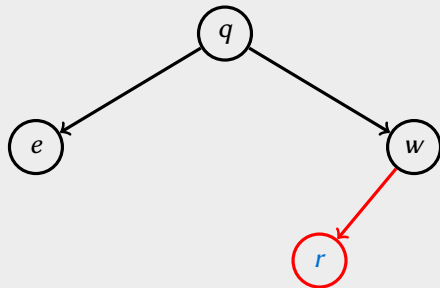
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Cambio de color en la raíz.



Árboles rojinegros

Ejemplo de inserción

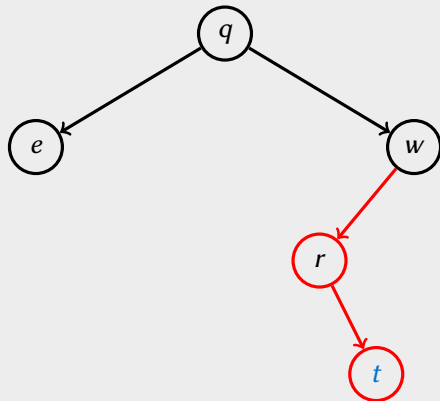
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja



Árboles rojinegros

Ejemplo de inserción

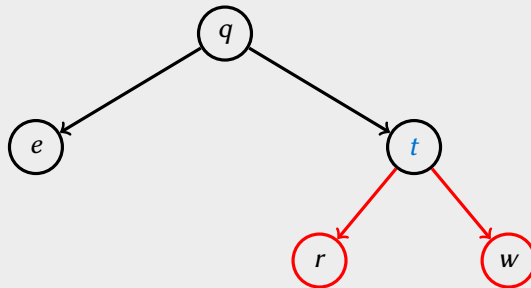
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Izquierda derecha.



Árboles rojinegros

Ejemplo de inserción

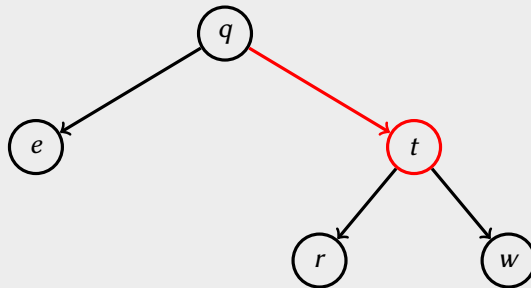
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Rotación doble izquierda derecha.



Árboles rojinegros

Ejemplo de inserción

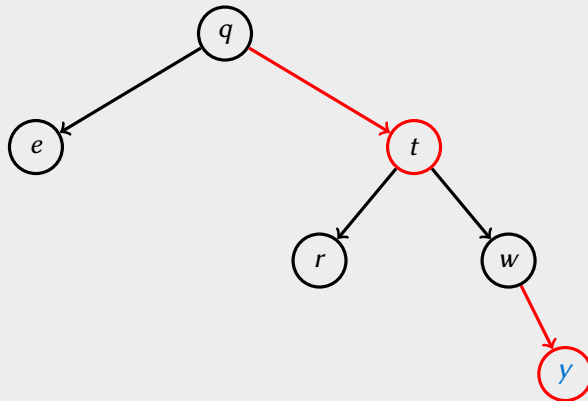
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Cambio de color en nodo t .



Árboles rojinegros

Ejemplo de inserción

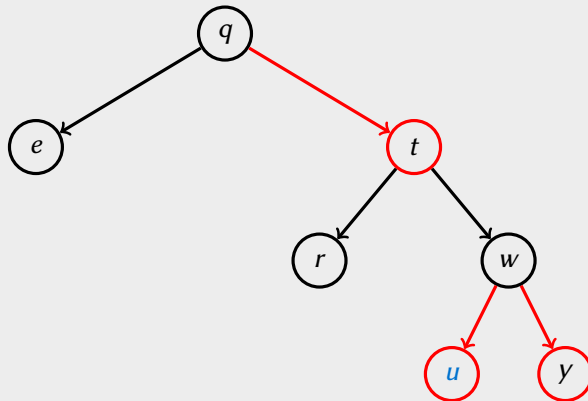
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja.



Árboles rojinegros

Ejemplo de inserción

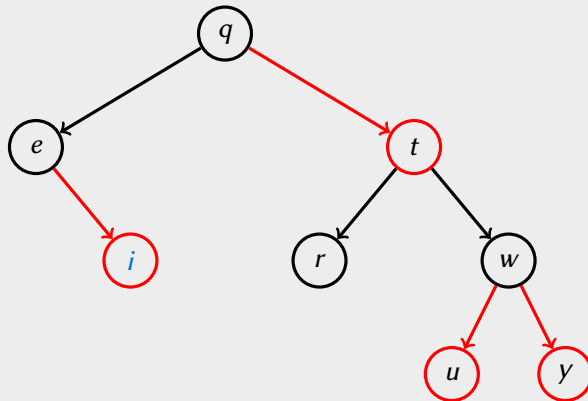
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja.



Árboles rojinegros

Ejemplo de inserción

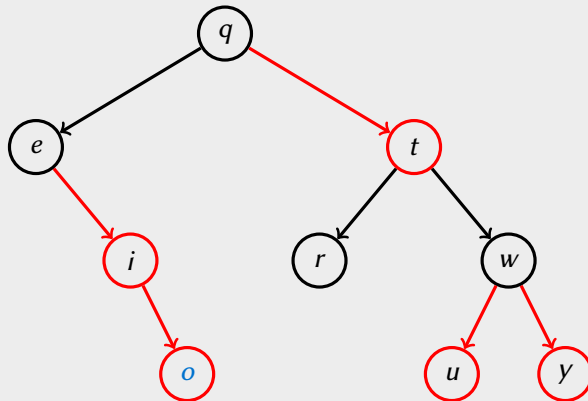
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja.



Árboles rojinegros

Ejemplo de inserción

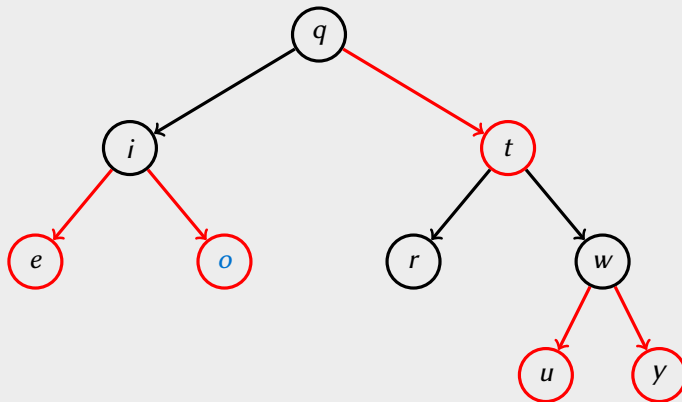
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Derecha derecha.



Árboles rojinegros

Ejemplo de inserción

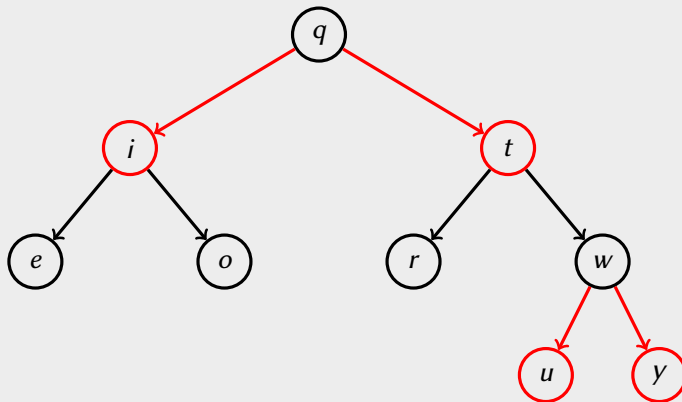
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Rotación simple derecha derecha.



Árboles rojinegros

Ejemplo de inserción

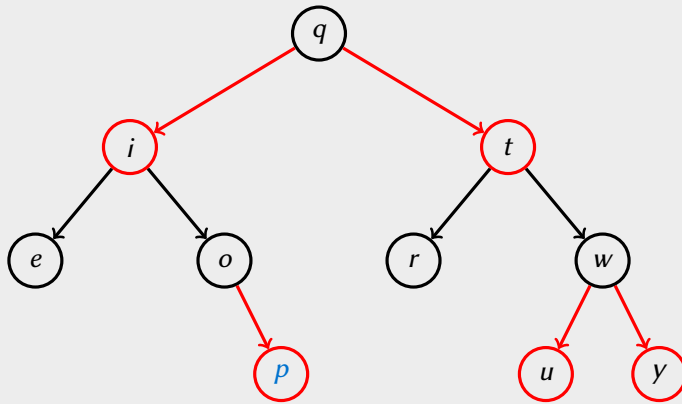
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Cambio de color en nodo i .



Árboles rojinegros

Ejemplo de inserción

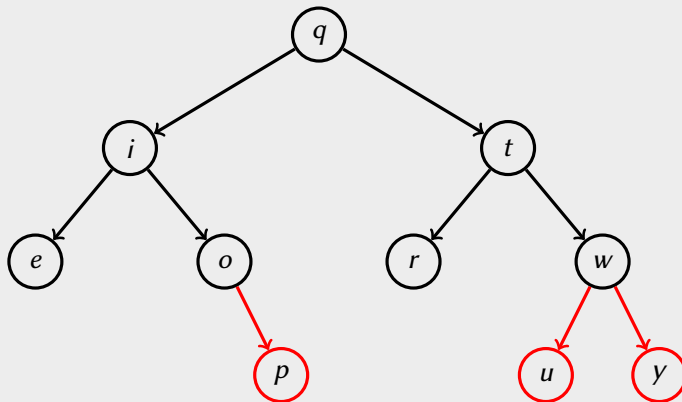
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja.



Árboles rojinegros

Ejemplo de inserción

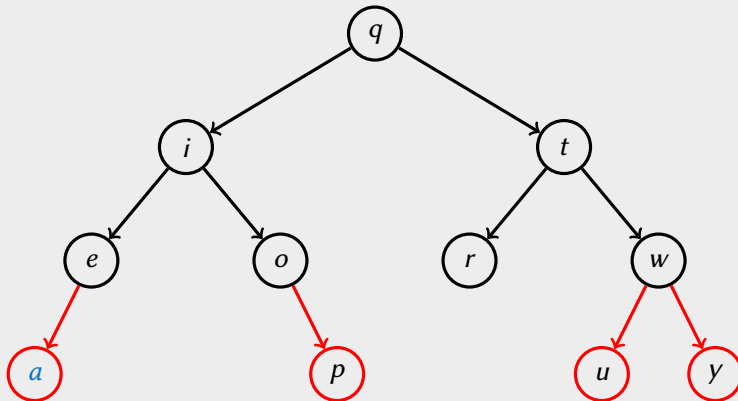
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Cambio de color en la raíz.



Árboles rojinegros

Ejemplo de inserción

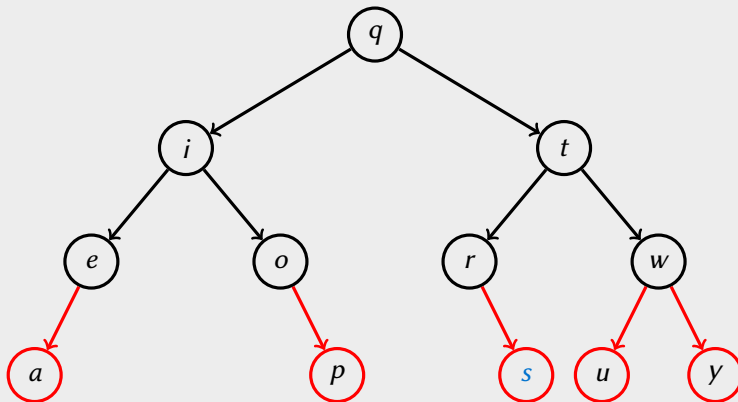
Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja.



Árboles rojinegros

Ejemplo de inserción

Claves en orden $q, w, e, r, t, y, u, i, o, p, a, s$. Hoja.



Ocho representaciones de conjuntos

Resumen de resultados

Número de pasos en el peor caso, sobre un conjunto A de n elementos y un elemento x .
Abajo $\log_2 n \leq h \leq n$ es la altura del árbol binario de búsqueda.

Operación	Símbolo	Bit	AD	AO	LD	LO	ABB	A234	ARN
crear	$\emptyset$	n	1	1	1	1	1	1	1
destruir		1	1	1	n	n	n	n	n
cardinalidad	$ A $	1	1	1	1	1	1	1	1
pertenencia	$x \in A$	1	n	$\log_2 n$	n	n	h	$\log_2 n$	$2 \log_2 n$
agregar	$A \cup x$	1	n	n	n	n	h	$\log_2 n$	$2 \log_2 n$
eliminar	$A \setminus x$	1	n	n	n	n	h	$\log_2 n$	$2 \log_2 n$

Objetivo cumplido: pertenencia, agregar y eliminar en $\approx \log_2 n$ pasos.