



Esta obra está bajo una Licencia Creative Commons Atribución 4.0



ESTRUCTURAS ARBORESCENTES (V 2.5)

Prof. José Fager – Paysandú agosto de 2017

fic

IPAD I

A horizontal bar at the top of the page, divided into a red section on the left and a teal section on the right.


Árboles

Definiciones

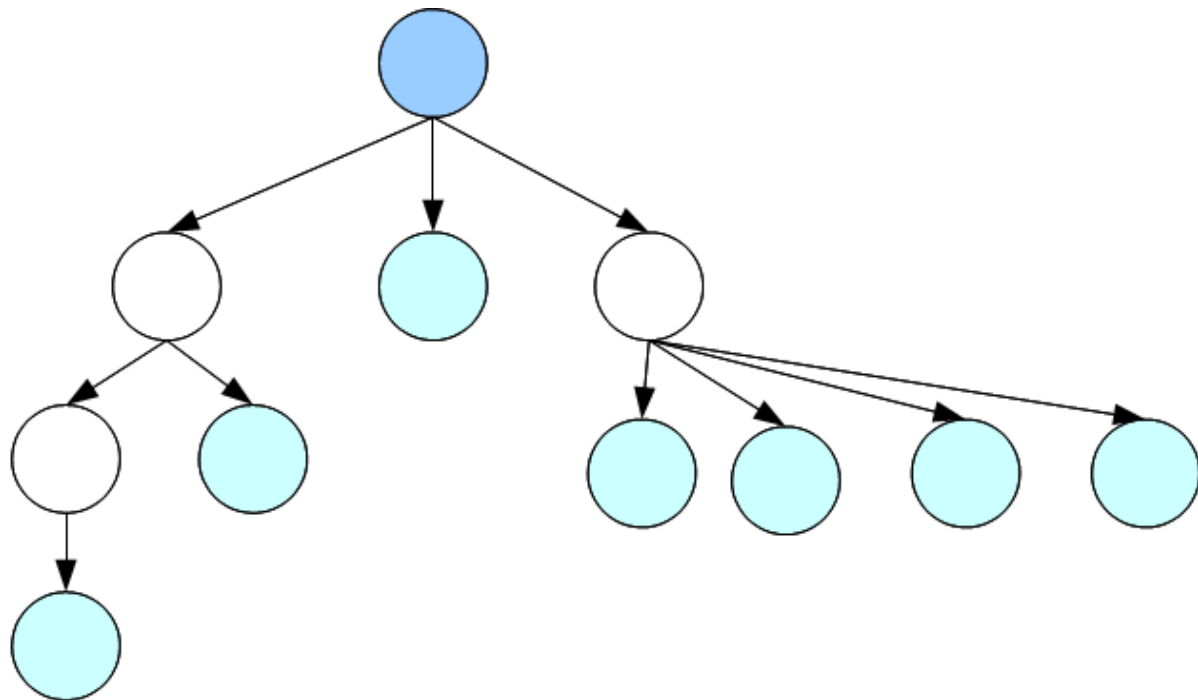
- ▣ Los árboles son estructuras que almacenan datos de forma arborescente.
- ▣ Los árboles se conforman por nodos.
- ▣ Los nodos se vinculan con otros nodos, según como sea la vinculación, se dice que unos son padres o hijos de los otros.
- ▣ Los nodos que no tienen hijos se les llaman hojas.
- ▣ El nodo que no tienen padres se lo llama raíz.
- ▣ Los nodos que no son hojas o raíz se llaman internos.
- ▣ Los árboles n-arios son árboles que sus nodos pueden tener hasta N hijos.

Ejemplo

raíz 

hojas 

internos 



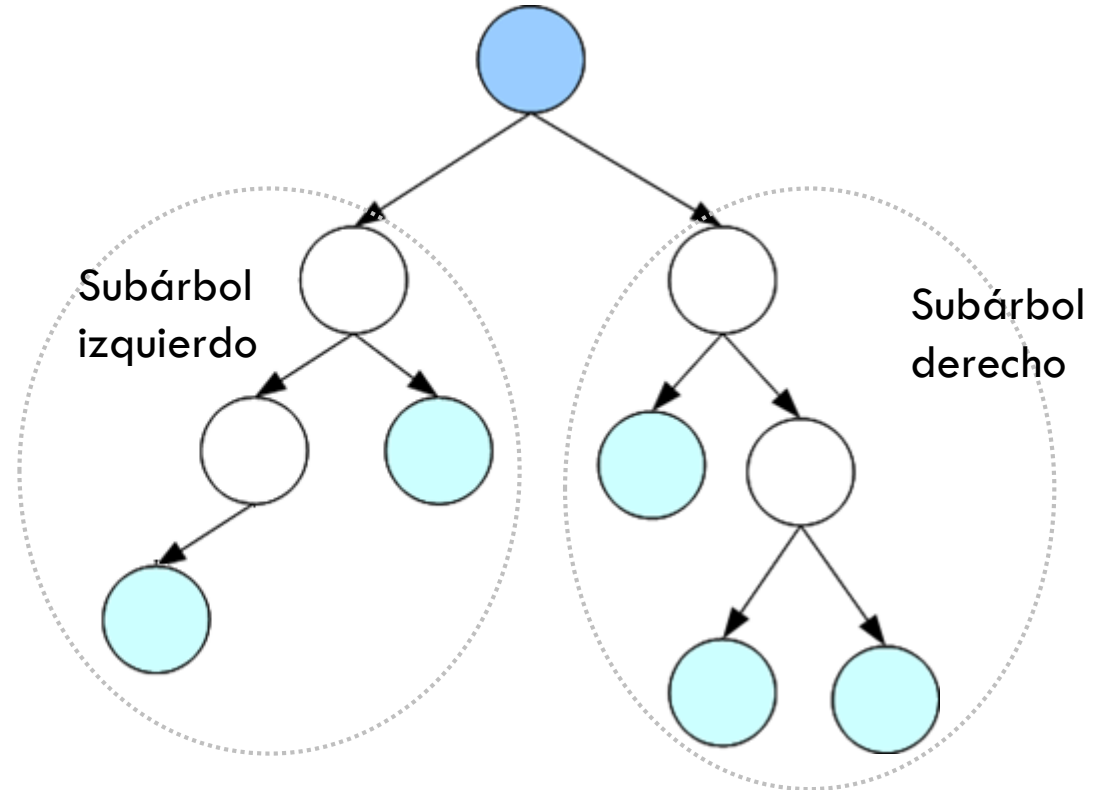
A horizontal decorative bar at the top of the slide, consisting of a red rectangular section on the left and a teal rectangular section on the right.

Árboles Binarios

Definiciones

- ▣ Los arboles binarios son árboles n-arios donde el valor de $N=2$.
- ▣ Los nodos de los árboles binarios pueden tener 2, 1 o 0 hijo.
- ▣ La estructura a la derecha de un nodo se llama “subárbol derecho”.
- ▣ La estructura a la izquierda de un nodo se llama “subárbol izquierdo”.

Ejemplo



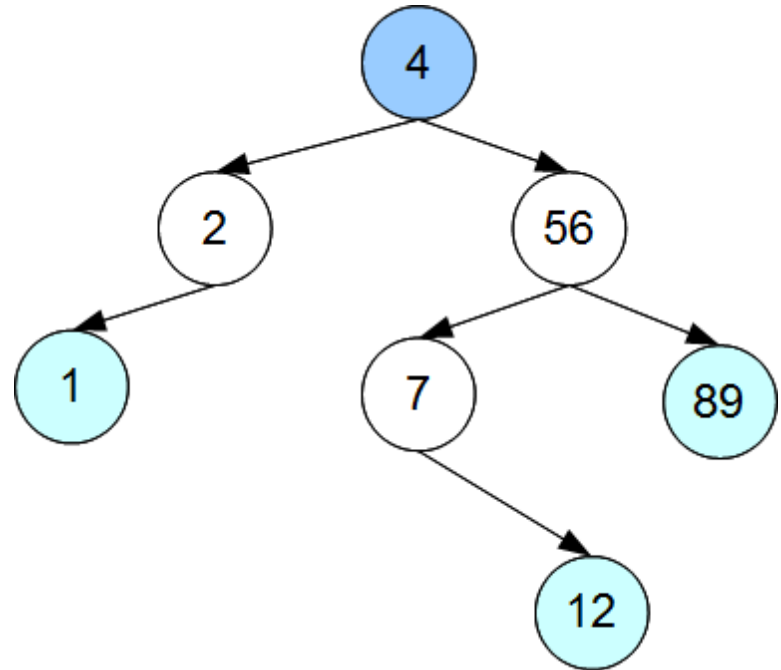
Árboles Binarios de Búsqueda (ABB)

Definición

- Los ABB son árboles binarios en donde los datos que se insertan en ellos guardan una relación particular. La definición formal de un ABB es:
 - 1) Todo árbol vacío es un árbol binario de búsqueda.
 - 2) Un árbol binario no vacío, de raíz R , es un árbol binario de búsqueda si:
 - 2.a) En caso de tener subárbol izquierdo, la raíz R debe ser mayor que el valor máximo almacenado en el subárbol izquierdo, y que el subárbol izquierdo sea un árbol binario de búsqueda.
 - 2.b) En caso de tener subárbol derecho, la raíz R debe ser menor que el valor mínimo almacenado en el subárbol derecho, y que el subárbol derecho sea un árbol binario de búsqueda.

ABB - ejemplo

- Si tomamos cualquier nodo del ABB, podremos observar que:
 - su subárbol izquierdo es vacío o todos sus nodos son menores que él,
 - su subárbol derecho es vacío o todos sus nodos son mayores que él.
- Esta es la principal propiedad que cumplen los ABB.



A horizontal bar at the top of the slide, divided into a red section on the left and a teal section on the right.

ABB Insertar datos

ABB - insertar datos

- ▣ La inserción de datos debe ser hecha de forma que el árbol siga siendo un ABB.
- ▣ El orden en que se insertan los datos condicionan la morfología del ABB.
- ▣ Si se quiere insertar un dato, se procede de la siguiente manera:
 - Si el ABB es vacío se crea el espacio para el nuevo elemento y se inserta.
 - Si no es vacío se compara el valor a insertar con el valor de la raíz:
 - Si el valor es menor que la raíz, se procede a insertarlo en su subárbol izquierdo.
 - Si el valor es mayor que la raíz, se procede a insertarlo en su subárbol derecho.
 - Si el valor es igual que la raíz, no se inserta.

ABB - insertar datos

- ▣ Se muestra un ejemplo partiendo de un ABB vacío de como se insertan datos.
- ▣ Suponemos que queremos ingresar los datos: 4, 56, 7, 89, 2, 1, y 12 en ese orden.

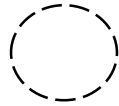
ABB - insertar el "4" y el "56"

El ABB está vacío.

ABB vacío



Se crea el espacio.



Se inserta el dato en el ABB.



Ahora como no está vacío se compara el valor del dato a insertar con el de la raíz, como $56 > 4$, se debe insertar en el subárbol derecho. Como el subárbol derecho está vacío se crea el espacio para el valor y se inserta.

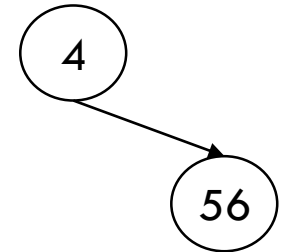
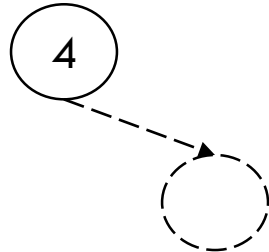


ABB - insertar el "7"

Como no está vacío se compara el valor del dato a insertar con el de la raíz, como $7 > 4$, se debe insertar en el subárbol derecho.

Como el subárbol derecho no está vacío se compara el valor del dato a insertar con el de la raíz del subárbol, como $7 < 56$, se debe insertar en el subárbol izquierdo de "56".

Como el subárbol izquierdo de "56" está vacío se crea el espacio para el valor y se inserta.

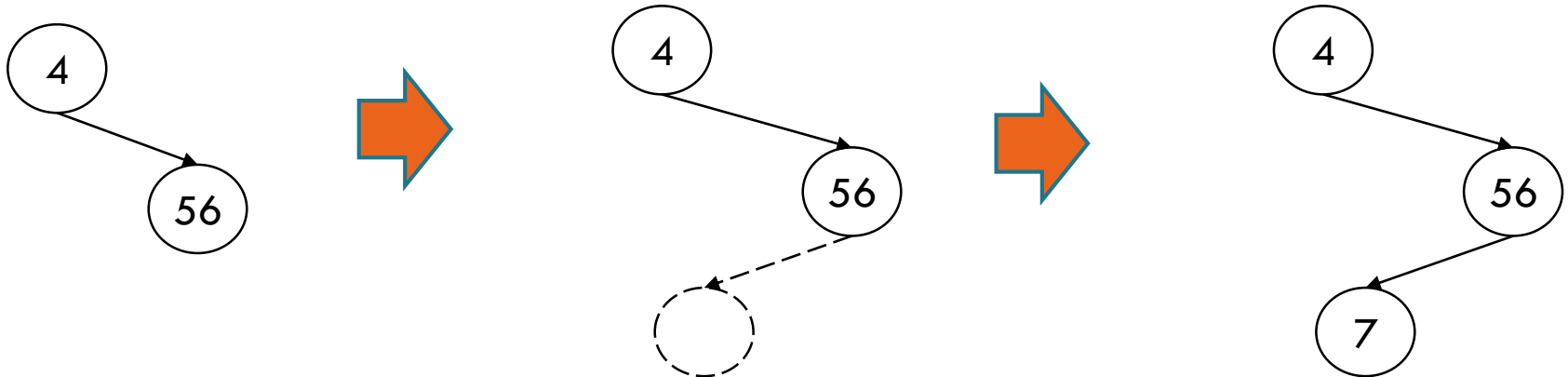


ABB - insertar el "89"

Como no está vacío se compara el valor del dato a insertar con el de la raíz, como $89 > 4$, se debe insertar en el subárbol derecho.

Como el subárbol derecho no está vacío se compara el valor del dato a insertar con el de la raíz del subárbol, como $89 > 56$, se debe insertar en el subárbol derecho de "56".

Como el subárbol derecho de "56" está vacío se crea el espacio para el valor y se inserta.

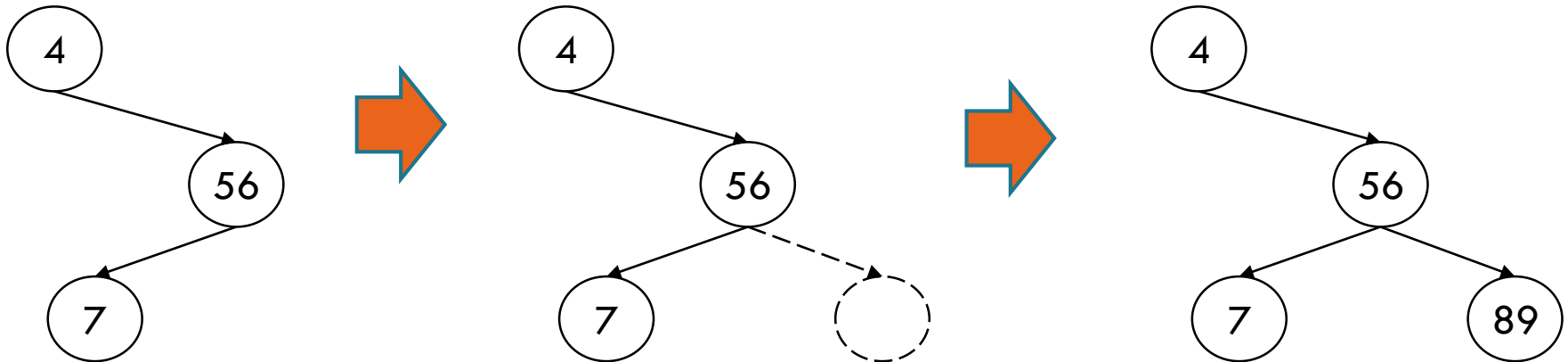


ABB - insertar el "2"

Como no está vacío se compara el valor del dato a insertar con el de la raíz, como $2 < 4$, se debe insertar en el subárbol izquierdo.

Como el subárbol izquierdo está vacío se crea el espacio para el valor y se inserta.

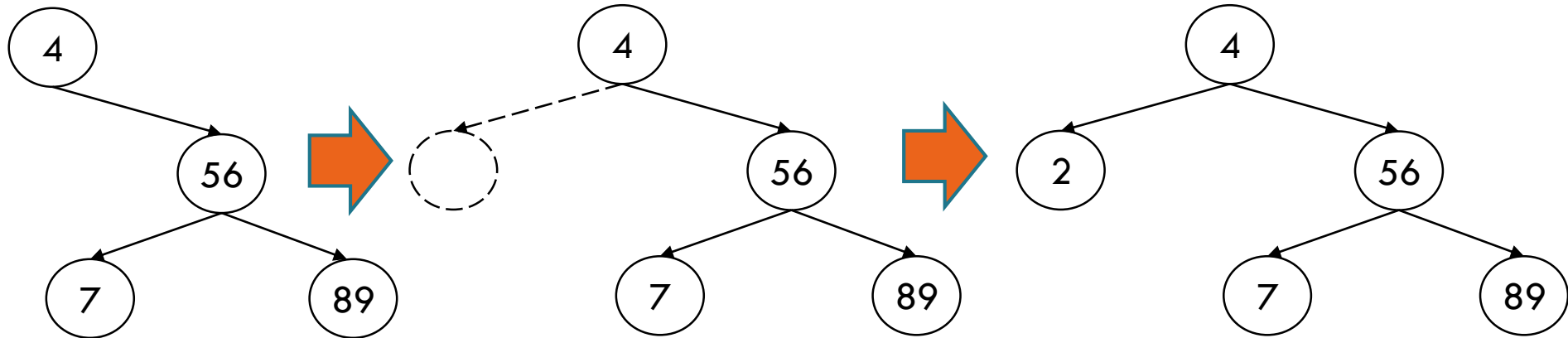


ABB - insertar el "1"

Análogamente a los casos anteriores, se compara el valor del dato a insertar con el de la raíz, como $1 < 4$, se debe insertar en el subárbol izquierdo.

Como el subárbol izquierdo no está vacío se compara el valor del dato a insertar con el de la raíz del subárbol, como $1 < 2$, se debe insertar en el subárbol izquierdo de "2".

Como el subárbol izquierdo de "2" está vacío se crea el espacio y se inserta.

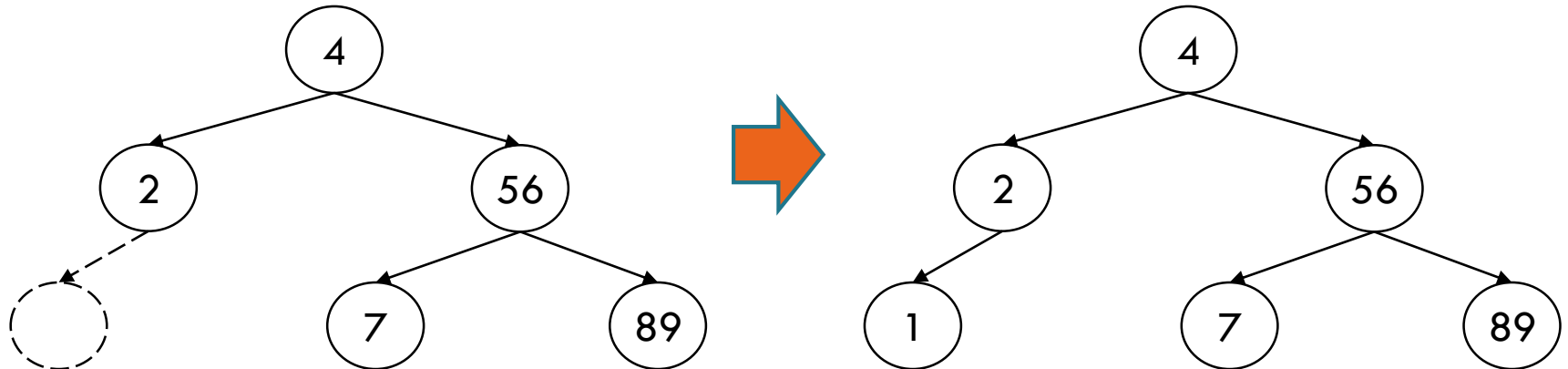
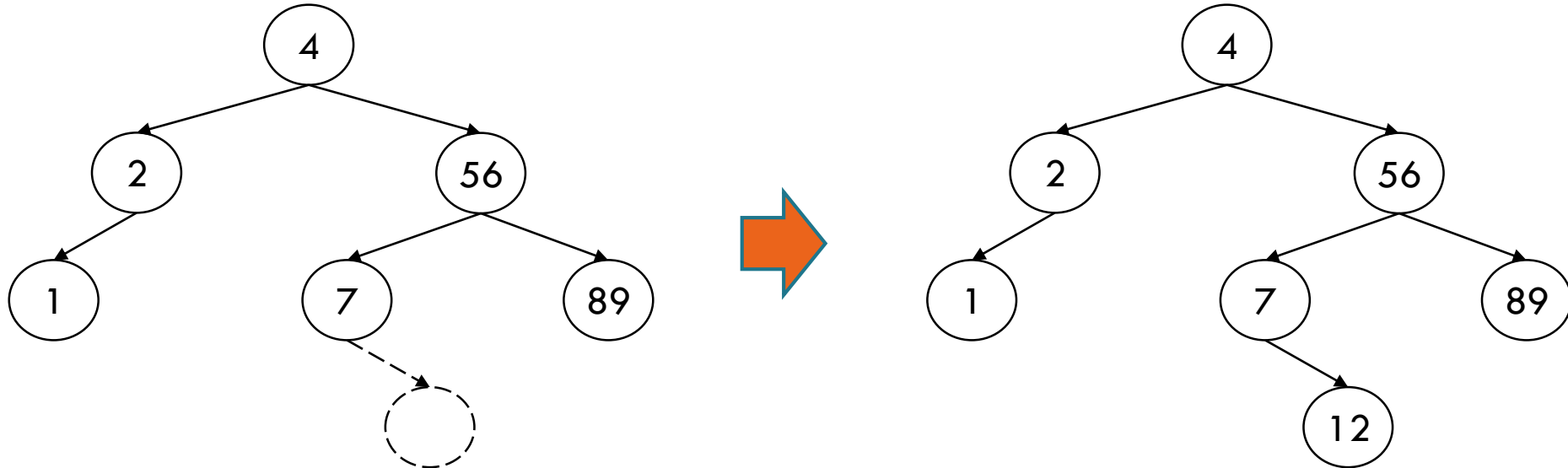


ABB - insertar el "12"

Análogamente a los casos anteriores, se compara el valor del dato a insertar con el de la raíz, como $12 > 4$, se debe insertar en el subárbol derecho, luego como $12 < 56$ se debe insertar en el subárbol izquierdo de "56", luego como $12 > 7$ se debe insertar en el subárbol derecho de "7", como el subárbol derecho de "7" está vacío se crea el espacio y se inserta.



A horizontal bar at the top of the page, divided into a red section on the left and a teal section on the right.

ABB Borrar datos

ABB - borrar datos

- Se muestran ejemplo partiendo de un ABB no vacío, para enseñar a borrar un dato QUE PERTENECE al árbol.
- Si se quiere borrar un dato que NO pertenece al árbol, el árbol no cambia.
- Hay 2 casos para el borrado:
 - El nodo a borrar ES UNA HOJA (sus dos subárboles son vacíos).
 - El nodo a borrar NO ES UNA HOJA (es un nodo interno o raíz).

ABB – Borrado “ES UNA HOJA”

- El nodo a borrar ES UNA HOJA (sus dos subárboles son vacíos)

ABB – borrar el “12” (ES UNA HOJA)

Si es una hoja el nodo a borrar simplemente se desconecta del ABB. Notar que el árbol sigue cumpliendo ser un ABB.

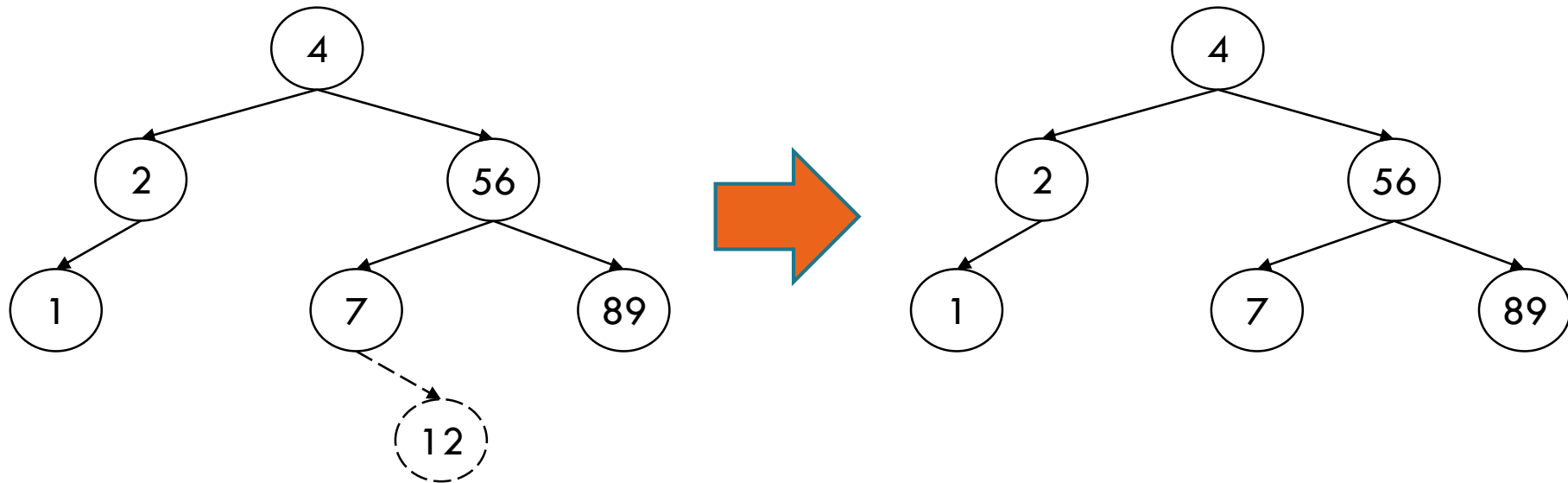


ABB – Borrado “NO ES UNA HOJA”



- El nodo a borrar NO ES UNA HOJA (es un nodo interno o raíz)

ABB – borrar el “4” (NO ES UNA HOJA)

Este es el caso más difícil, existen varios métodos, veremos el método de “intercambio”.

El método de “intercambio” consiste en ir intercambiando el nodo a borrar con:

- el menor nodo de su subárbol derecho o
- el mayor nodo de su subárbol izquierdo sucesivas veces, hasta que el nodo a borrar se “transforme” en una hoja.

Se debe tener en cuenta que en varios momentos del proceso el árbol que tendremos NO será un ABB!!!

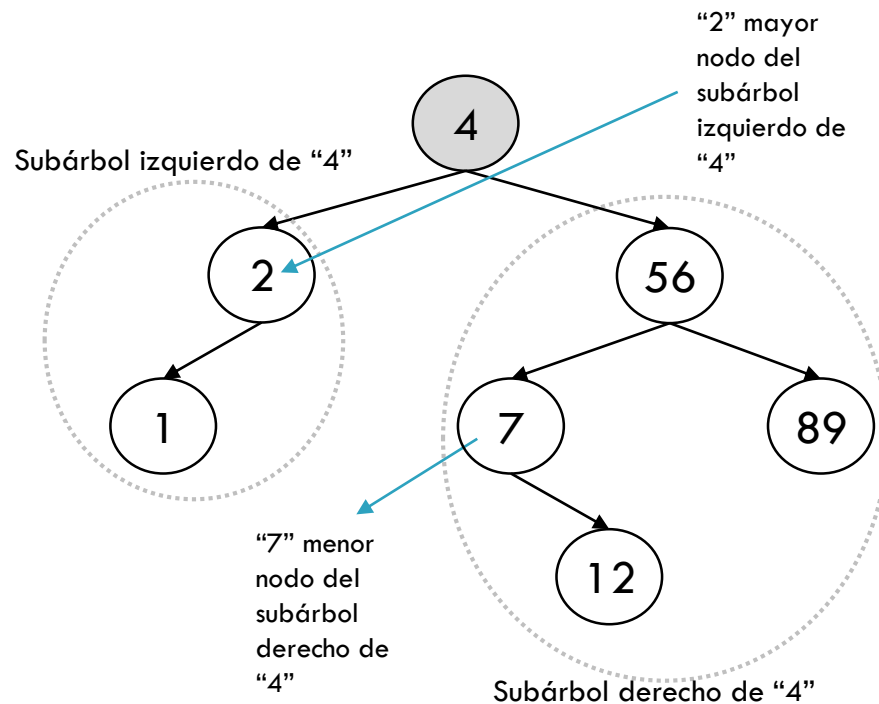
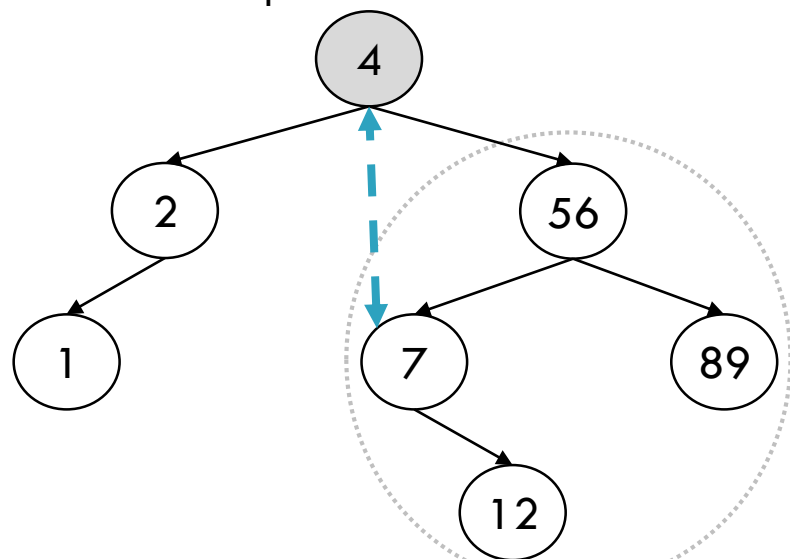
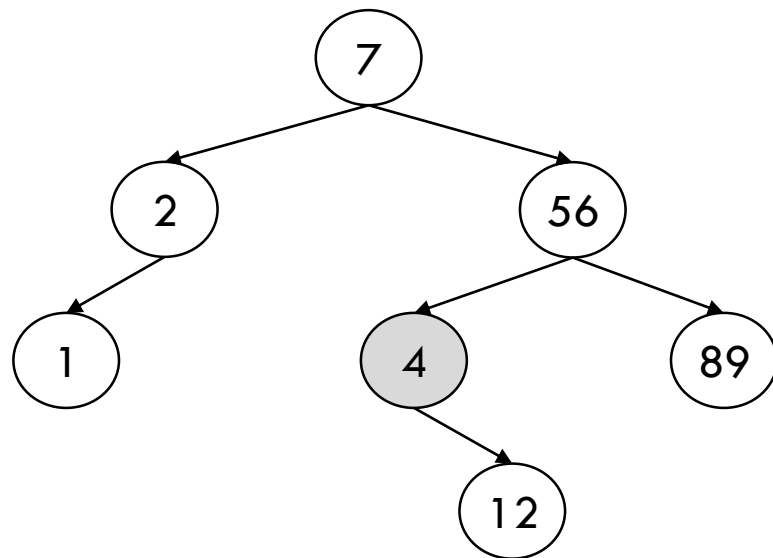


ABB – borrar el “4” (NO ES UNA HOJA)

Como “7” es el menor nodo del subárbol derecho y “2” el mayor de su subárbol izquierdo, del nodo a borrar (4), se puede intercambiar el “4” por “7” o el “4” por el “2”. En el ejemplo se realiza el primer caso.



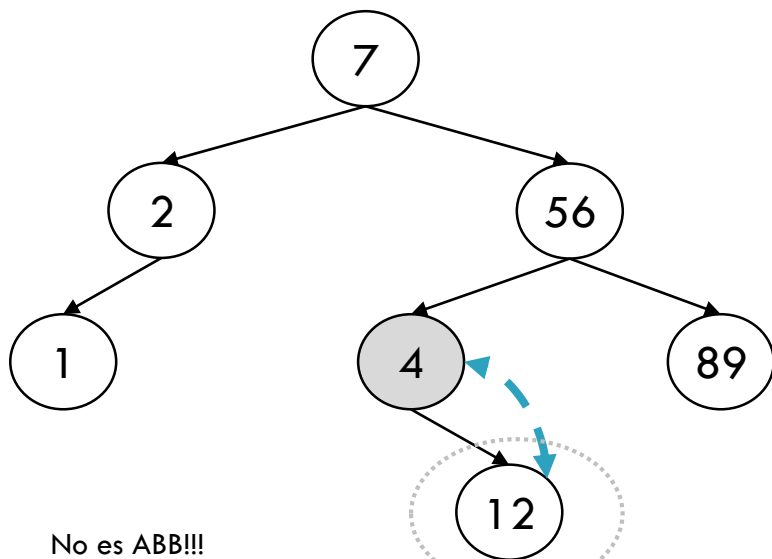
Subárbol derecho de “4”



Aún no es ABB!!!

ABB – borrar el “4” (NO ES UNA HOJA)

Como “4” aún no es una hoja, se toma “12” que es el menor nodo del subárbol derecho del nodo a borrar (4), se intercambia el “4” con el “12” (la opción del subárbol izquierdo no se considera por estar vacío).



Subárbol derecho de “4”

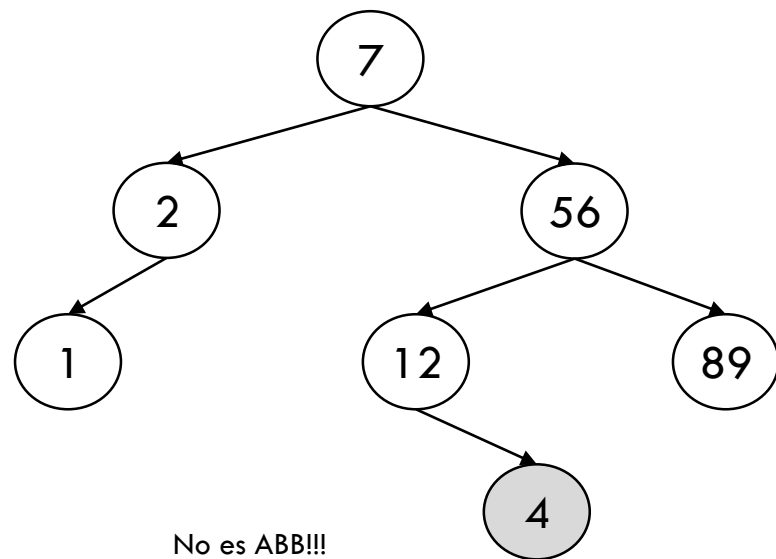
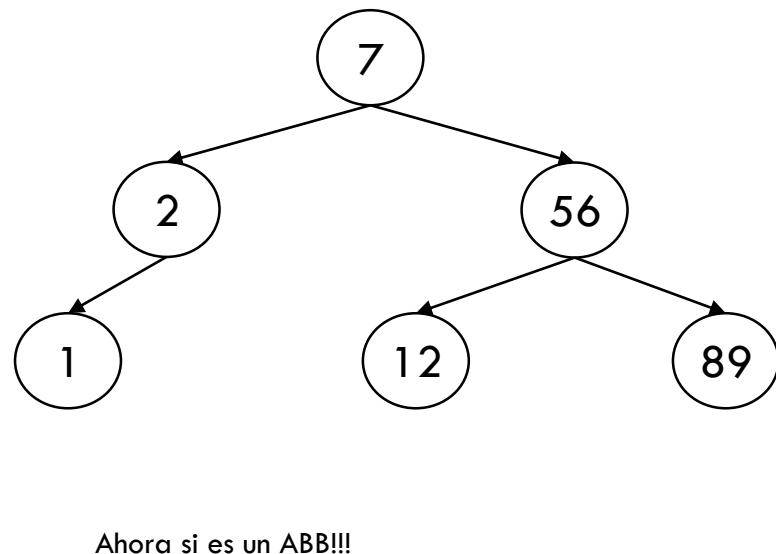
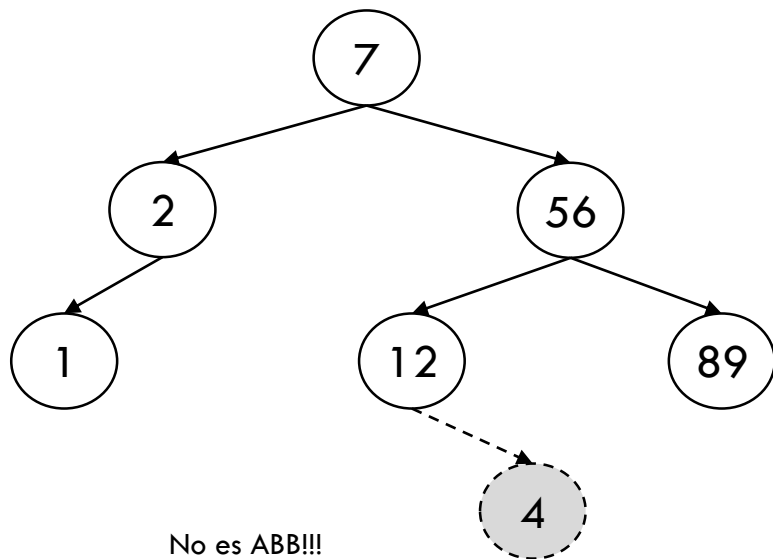


ABB – borrar el “4” (NO ES UNA HOJA)

Como ahora “4” si es una hoja, se lo borra como se haría con una hoja.



A horizontal bar at the top of the page, divided into a red section on the left and a teal section on the right.

TAD ABB

Operaciones del TAD ABB

//*** CONSTRUCTORAS ***//

//Retorna el ABB vacío

CrearABB() → ABB

//Inserta el elemento “e” en el ABB “a”

//Si “e” ya pertenece a “a” no lo inserta

AgregarABB(a, e)

//Quita el elemento “e” del ABB “a”

//Si “e” no pertenece a “a” no modifica “a”

BorrarABB(a, e)

//*** PREDICADO ***//

//Retorna true si el ABB “a” es vacío sino retorna false

EsVacioABB(a) → bool

//Retorna true si el elemento “e” pertenece al ABB “a” sino retorna false

PerteneceABB(a, e) → bool

Operaciones del TAD ABB

// *** SELECTORAS ***//

//Retorna el subárbol izquierdo del ABB "a".

// Si "a" es vacío retorna "a"

IzqABB(a) → ABB

//Retorna el subárbol derecho del ABB "a".

//Si "a" es vacío retorna "a"

DerABB(a) → ABB

//Retorna el mínimo valor del ABB "a"

// Si "a" es vacío retorna el elemento vacío

MinABB(a) → element

//Retorna el máximo valor del ABB "a"

// Si "a" es vacío retorna el elemento vacío

MaxABB(a) → element

//Retorna la raíz del ABB "a"

// Si "a" es vacío retorna el elemento vacío

RaizABB(a) → element

A horizontal bar at the top of the page, divided into a red section on the left and a teal section on the right.

TAD Diccionario

Definición

- El TAD Diccionario es un ejemplo de TAD que se puede implementar usando ABB.
- Sus operaciones básicas son:
 - Crear el diccionario vacío
 - Insertar un elemento en el diccionario.
 - Quitar un elemento del diccionario.
 - Preguntar si un elemento pertenece o no al diccionario.

Operaciones del TAD Diccionario

//*** CONSTRUCTORAS ***//

//Retorna el Diccionario vacío

Nuevo() → Diccionario

//Inserta el elemento “e” en el Diccionario “dic”

//Si “e” ya pertenece a “dic” no lo inserta

Insertar(dic, e)

//Quita el elemento “e” del Diccionario “dic”

//Si “e” no pertenece a “dic” no modifica “dic”

Borrar(a, e)

//*** PREDICADO ***//

//Retorna true si el Diccionario “dic” es vacío

//sino retorna false

EsVacio(dic)→ bool

//Retorna true si el elemento “e” pertenece al

//Diccionario “dic” sino retorna false

Pertenece(dic, e) → bool

Ejemplo – Corrector Ortográfico

- Un ejemplo es un corrector ortográfico simple para corregir un texto.
- El corrector por cada palabra del texto:
 - se fija si pertenece al diccionario:
 - sino pertenece da la opción de agregarla al diccionario
 - Si pertenece no hace nada
- Se puede implementar usando el TAD Diccionario