



ÁRBOL

ESTRUCTURAS DE DATOS
y ALGORITMOS
LCC – LSI - TUPW

Árbol Balanceado - AVL

Un árbol es *perfectamente equilibrado*, si para cada nodo los números de nodos en sus subárboles izquierdo y derecho difieren cuanto más en uno.

Un árbol está *balanceado* si y solo si en cada nodo las alturas de sus dos subárboles difieren a lo máximo en uno.

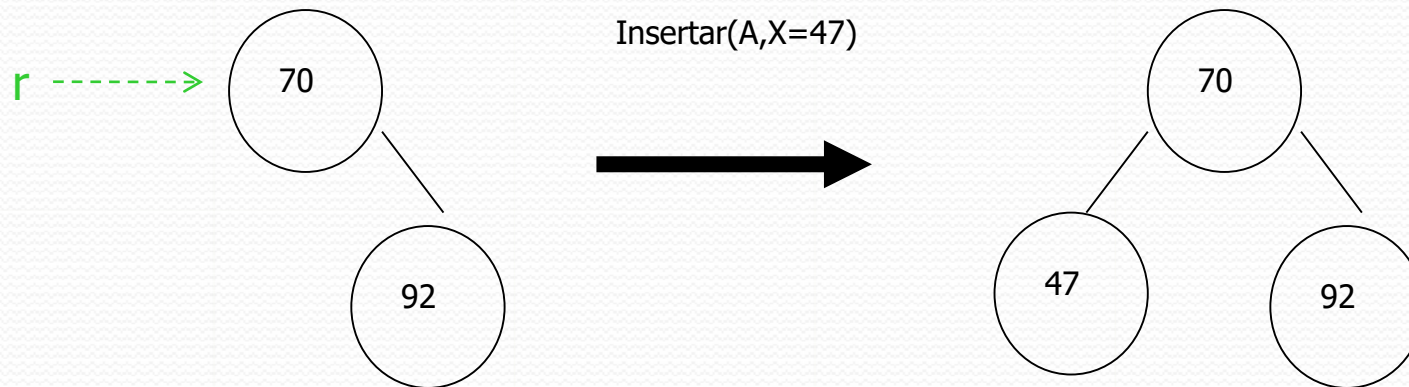
Los árboles balanceados reciben el nombre de **Árboles AVL** por ser Adelson-Velski y Landis quienes propusieron esta definición de equilibrio.

T.A.D. Arbol Balanceado

Construcción de operaciones abstractas (1)

Insertión en un Árbol Balanceado

a) Si $\text{altura}(I(r)) < \text{altura}(D(r))$ y X se inserta en $I(r)$

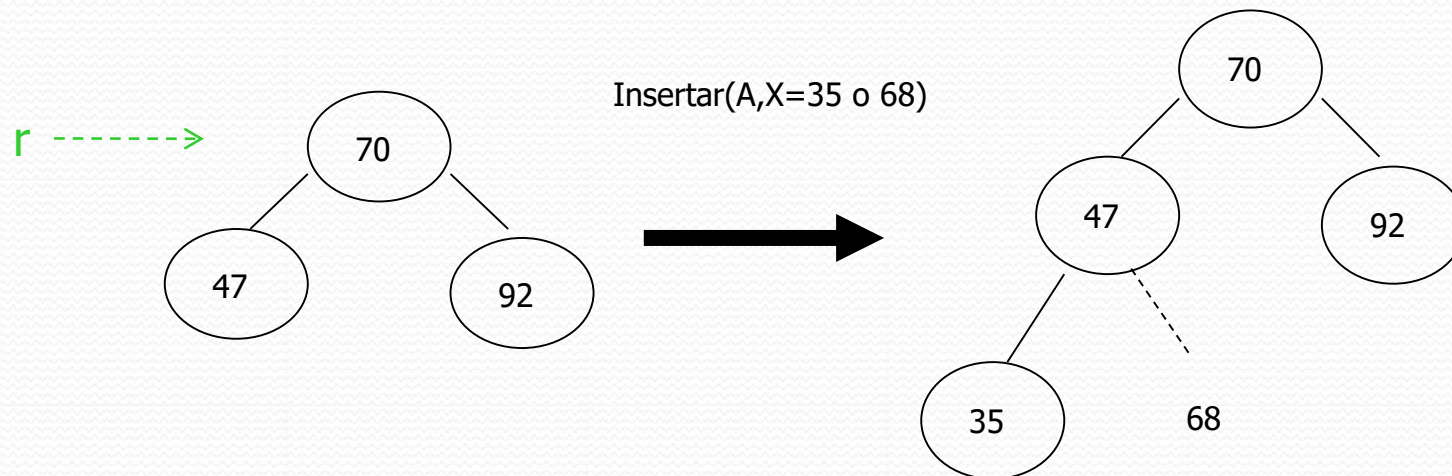


T.A.D. Arbol Balanceado

Construcción de operaciones abstractas (2)

Insertión en un Árbol Balanceado

b) Si $\text{altura}(\text{I}(r)) = \text{altura}(\text{D}(r))$ y X se inserta en $\text{I}(r)$,

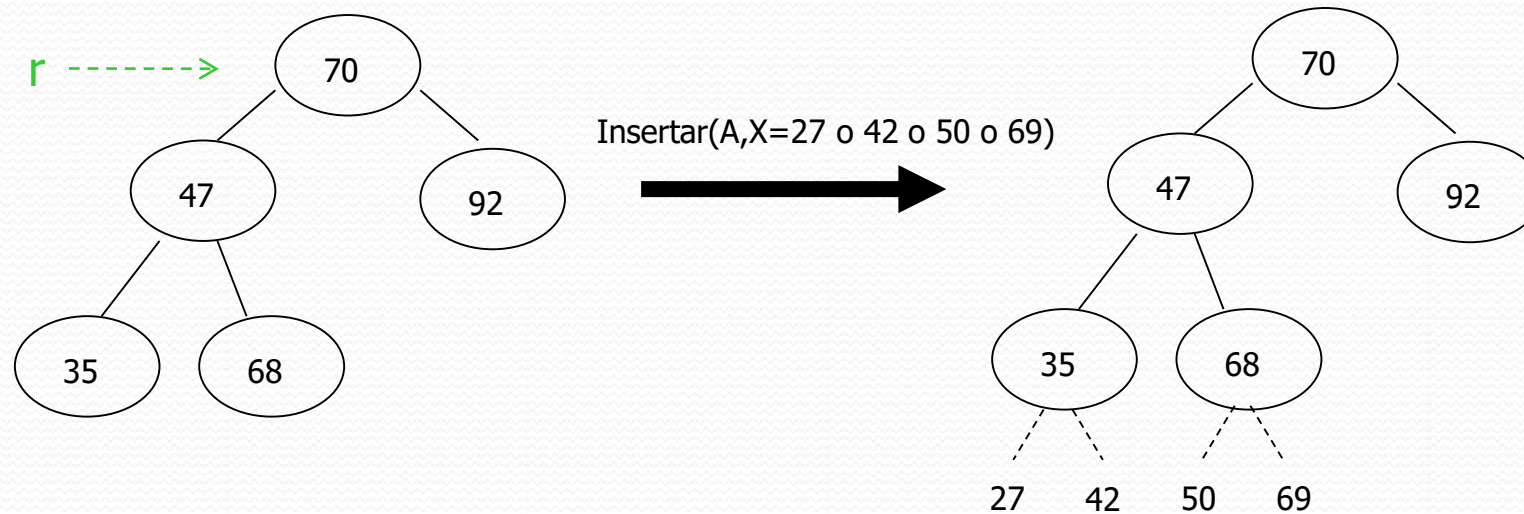


T.A.D. Arbol Balanceado

Construcción de operaciones abstractas (3)

Inserción en un Árbol Balanceado

c) Si $\text{altura}(I(r)) > \text{altura}(D(r))$ y X se inserta en $I(r)$



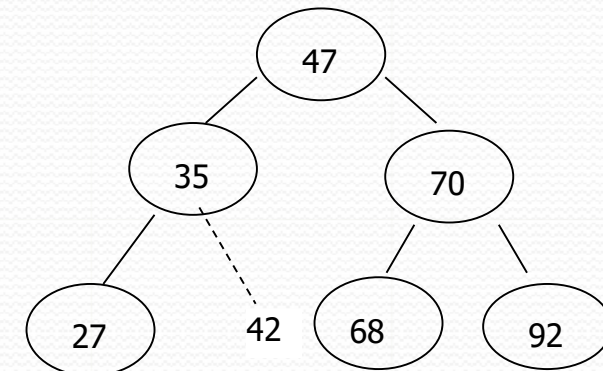
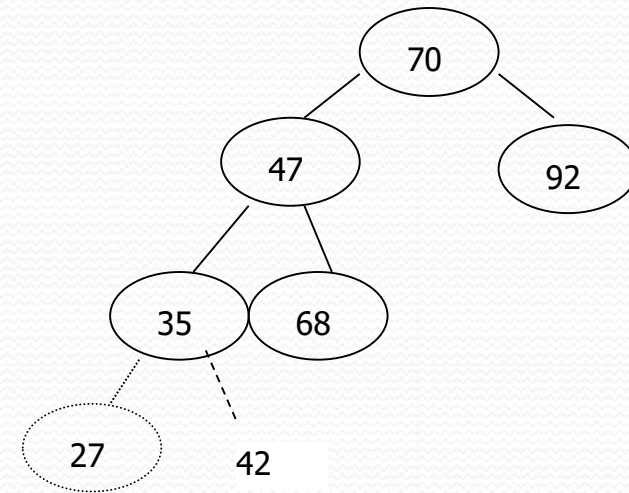
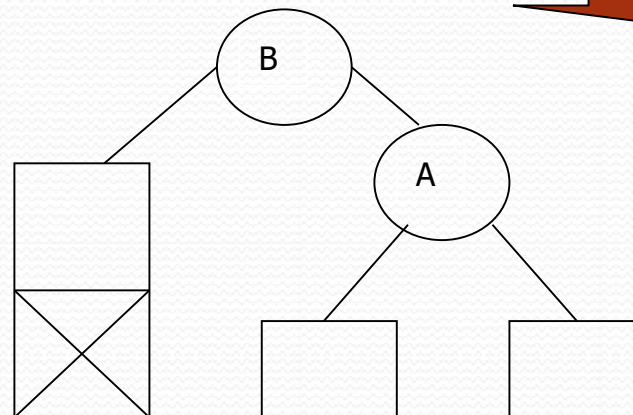
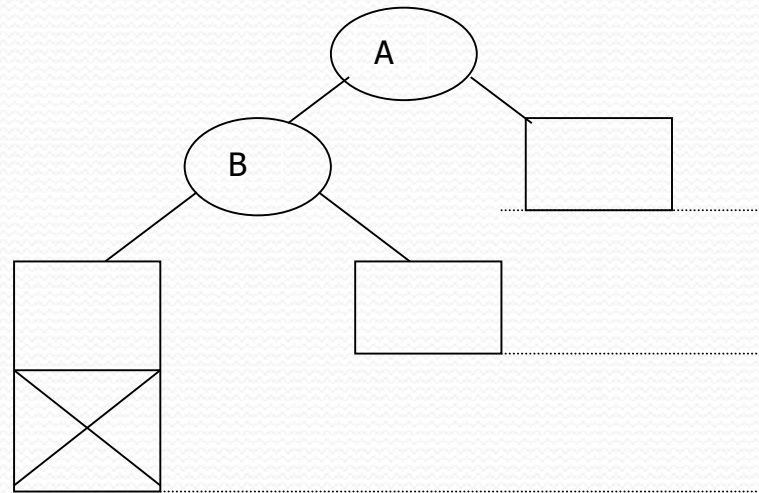
REBALANCEAR !!

T.A.D. Arbol Balanceado

Construcción de operaciones abstractas (4)

R
E
B
A
L
A
N
C
E
O

C
A
S
O
1

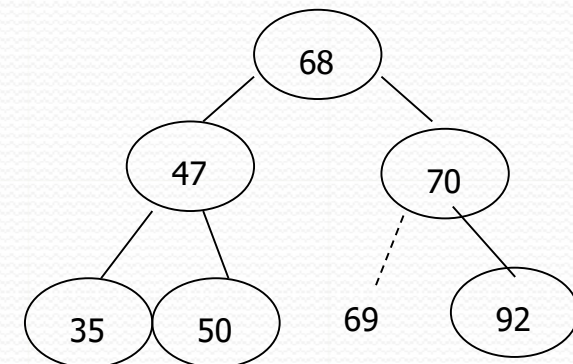
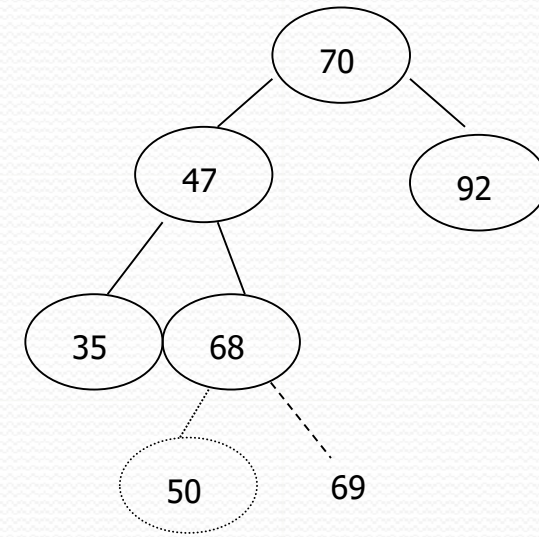
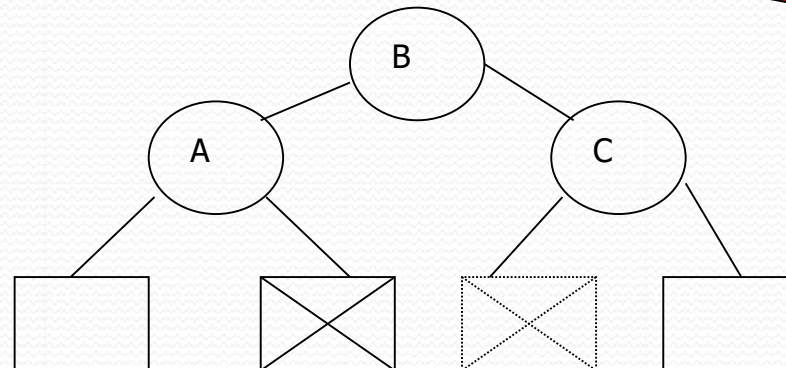
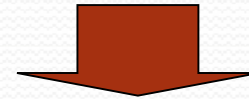
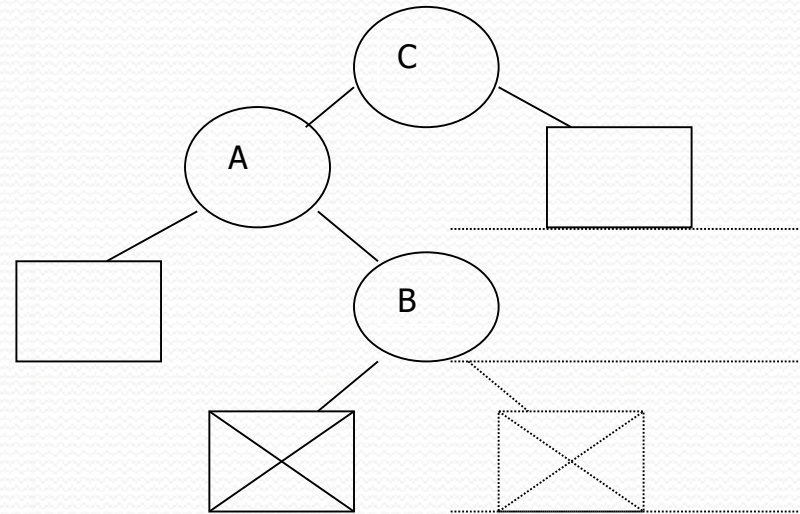


T.A.D. Arbol Balanceado

Construcción de operaciones abstractas (5)

R
E
B
A
L
A
N
C
E
O

C
A
S
O
2



Árbol Multicamino

Construcción y mantención de árboles de búsqueda a gran escala, que se almacenan en memoria secundaria.

Almacenar datos de 1 millón de elementos árbol balanceado $\rightarrow \text{Log}_2^{10^6} = 20$ comp.
acceso a disco por cada comp . 20 accesos a disco

Se incorpora un tipo particular de árbol multicamino

cantidad de accesos en el peor de los casos sería: $\text{Log}_{100}^{10^6} = 3$ accesos

Cada página (salvo una) contiene entre **n y $2n$** nodos para determinada constante **n** .
De ahí que, en un árbol con N elementos y un tamaño máximo de página de $2n$ nodos por página, en el peor caso requiere $\log_{2n} N$ accesos de página.

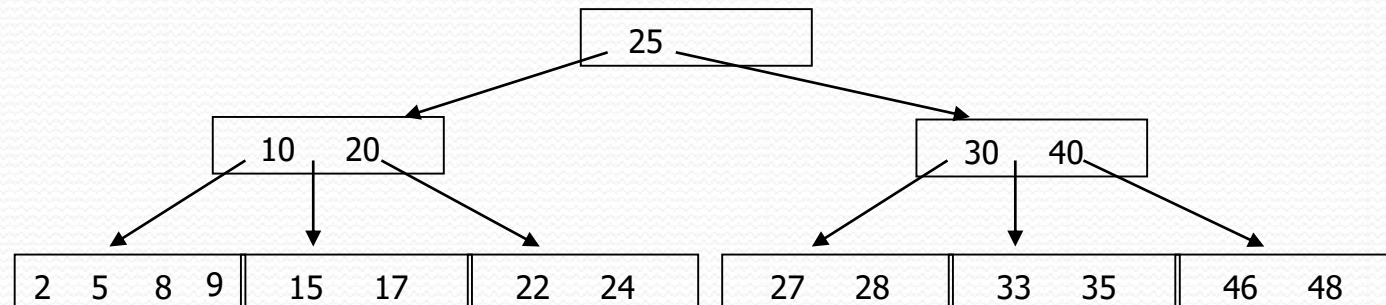
Árbol B, árbol multcamino de *orden n*:

- 1) Cada página contiene a lo sumo $2n$ elementos (claves).
- 2) Cada página, excepto la pagina raíz, contiene n elementos por lo menos.
- 3) Cada página es una página de hoja, o sea que no tiene descendientes, o tiene $m+1$ descendientes, donde m es su número de claves en esa página ($n \leq m \leq 2n$).
- 4) Todas las páginas hoja aparecen al mismo nivel.

T.A.D. Árbol B de orden 2

Especificación

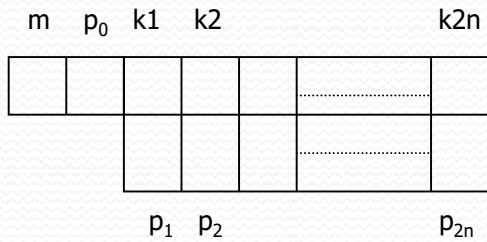
- 1) Cada página contiene a lo sumo 4 ($2*2$) elementos (claves).
- 2) Cada página, excepto la pagina raíz, contiene 2 elementos por lo menos.
- 3) Cada página es una página de hoja, o sea que no tiene descendientes, o tiene $m+1$ descendientes, donde m es su número de claves en esa página ($2 \leq m \leq 4$).
- 4) Todas las páginas hoja aparecen al mismo nivel.



T.A.D. Árbol B

Representación

Estructura de la Página



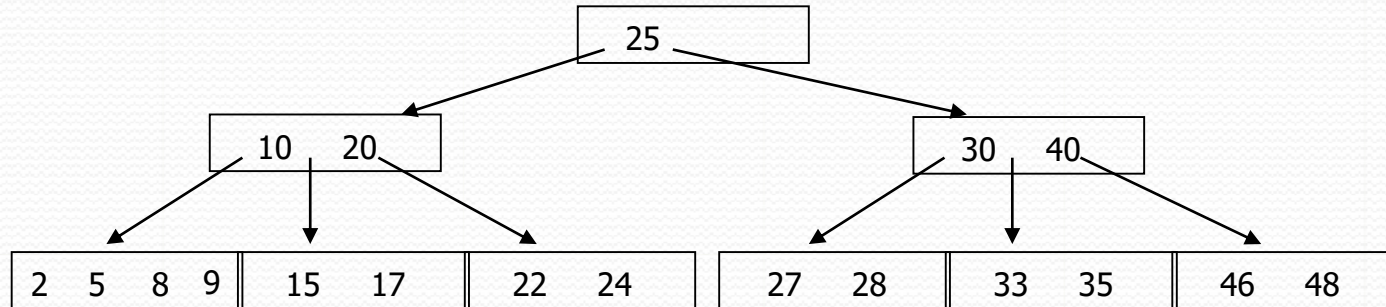
m : cantidad de claves en la página

$$k_i : \text{clave}; \quad 1 \leq i \leq m$$

p0 : dirección de la página que contiene claves menores que k1

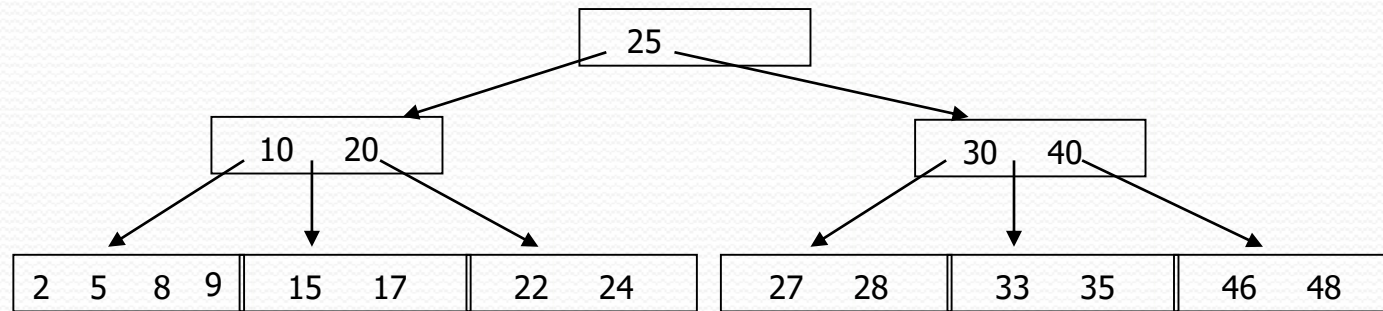
p_i : dirección de la página que contiene claves mayores que k_i y menores que k_{i+1}

pm : dirección de la página que contiene claves
mayores que km



T.A.D. Árbol B

Operaciones Abstractas: Buscar



¿Cómo se realiza la búsqueda de una clave **X**, en una página dada?

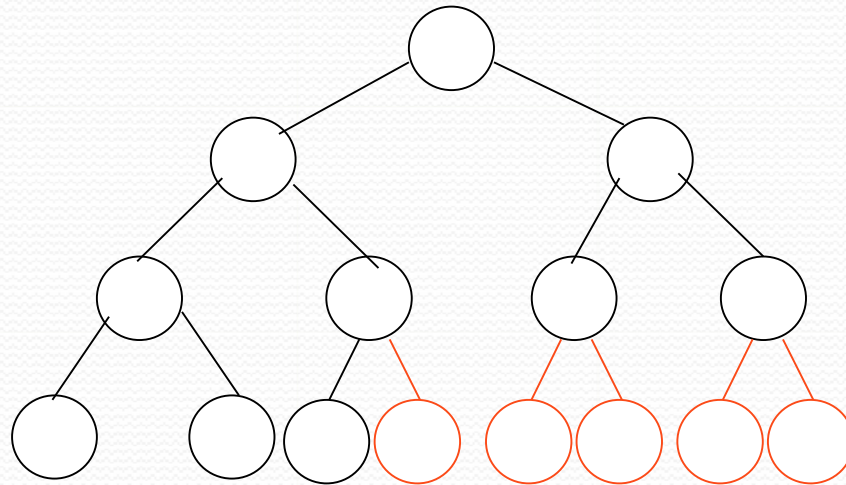
- 1) $k_m < \mathbf{X}$;
- 2) $\mathbf{X} < k_1$;
- 3) $k_i < \mathbf{X} < k_{i+1}$ ($1 \leq i < m$)

¿Si la clave **X** no se encuentra en una página, cómo continúa la búsqueda?

entonces la búsqueda continúa por la página apuntada por p_m
entonces la búsqueda continúa por la página apuntada por p_0
entonces la búsqueda continúa por la página apuntada por p_i

Árbol Binario Semicompleto

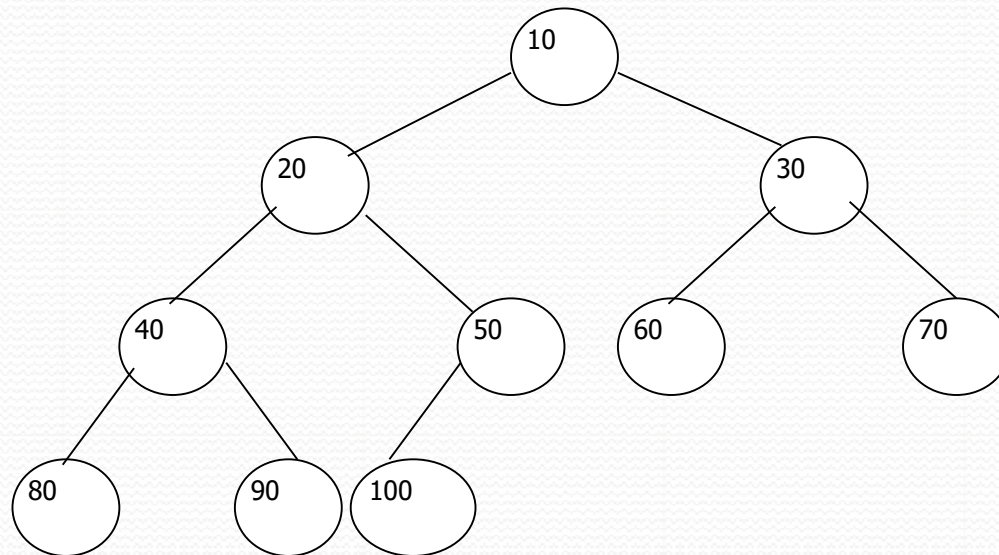
Árbol Binario Semicompleto: Un Árbol Binario Semicompleto de n nodos, se forma a partir de un árbol binario completo de $n+q$ nodos, quitando las q hojas extremo derechas del árbol binario completo.



T.A.D. Montículo Binario

Especificación

Montículo Binario: Un *Montículo Binario* es un árbol binario semicompleto en el que el valor de clave almacenado en cualquier nodo es menor o igual que el valor de clave de sus hijos.



Montículos Binarios

Colas de Prioridad



Los valores –claves- que representan prioridades deben interpretarse de la siguiente manera: **a menor valor-mayor prioridad**, por lo que **la máxima prioridad se encuentra en la raíz del árbol**.

T.A.D.Montículo Binario

Especificación

Operaciones Abstractas

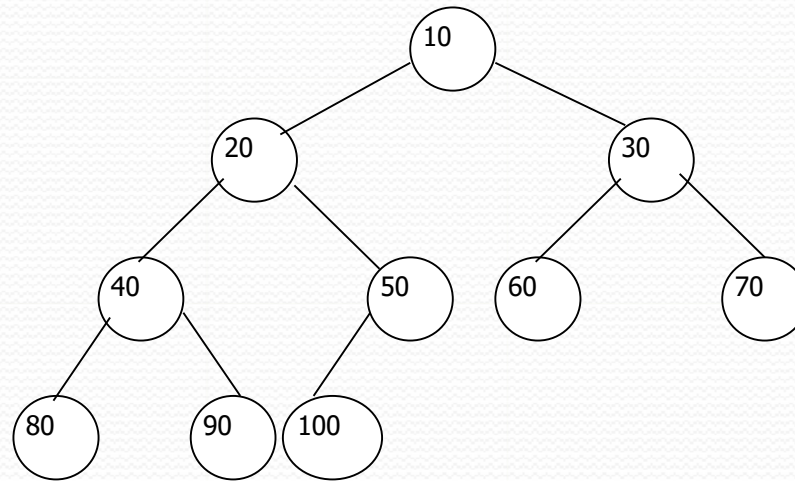
M: Montículo Binario y X: Clave

<i>NOMBRE</i>	<i>ENCABEZADO</i>	<i>FUNCIÓN</i>	<i>ENTRADA</i>	<i>SALIDA</i>
Insertar	Insertar(M, X)	Ingresa el elemento X al montículo M	M, X	M con el nuevo elemento
Eliminar_Mínimo	Eliminar_Mínimo(M, X)	Suprime del montículo M el elemento de máxima prioridad- mínimo valor de clave	M	M y X: elemento de máxima prioridad

T.A.D. Montículo Binario

Representación

Montículo Binario desde una perspectiva conceptual



Elementos

	10	20	30	40	50	60	70	80	90	100
0	1	2	3	4	5	6	7	8	9	10	

Elementos [i]

Padre : Elementos [$\lfloor i / 2 \rfloor$]

Hijo Izquierdo : Elementos [$i * 2$]

Hijo Derecho : Elementos [$i * 2 + 1$]

Montículo Binario en su almacenamiento

T.A.D. Montículo Binario

Construcción de operaciones abstractas (1)

Tanto la operación ***Insertar*** como ***Eliminar_Mínimo***, deben garantizar que en el Montículo Binario se mantengan las siguientes dos propiedades:

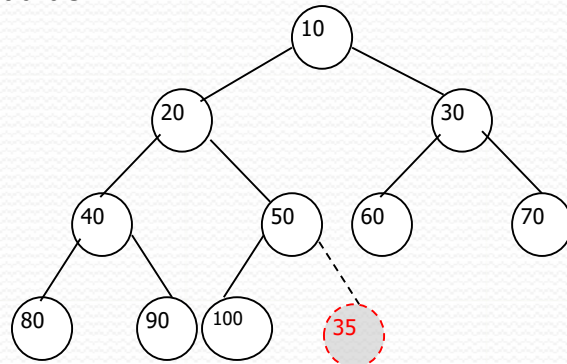
- ***Propiedad de Estructura*** : el objeto de datos debe ser un árbol binario semicompleto.
- ***Propiedad de Orden*** : el valor de clave almacenado en cualquier nodo debe ser menor o igual que el valor de clave de sus hijos.

T.A.D. Montículo Binario

Construcción de operaciones abstractas (2)

Insertar (M,X=35)

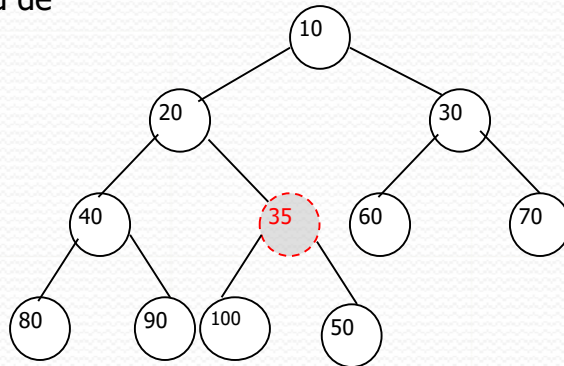
a) Propiedad de Estructura



Elementos

	10	20	30	40	50	60	70	80	90	100	35
0	1	2	3	4	5	6	7	8	9	10	11		

b) Propiedad de Orden



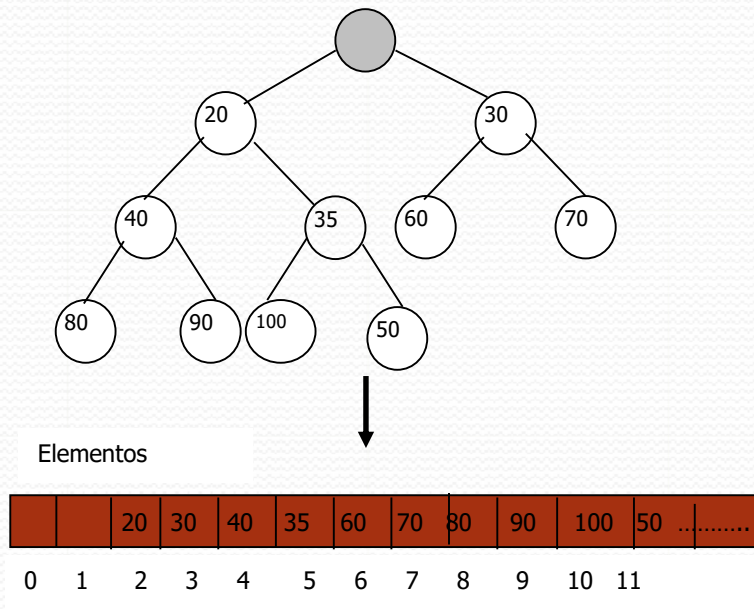
Elementos

	10	20	30	40	35	60	70	80	90	100	50
0	1	2	3	4	5	6	7	8	9	10	11		

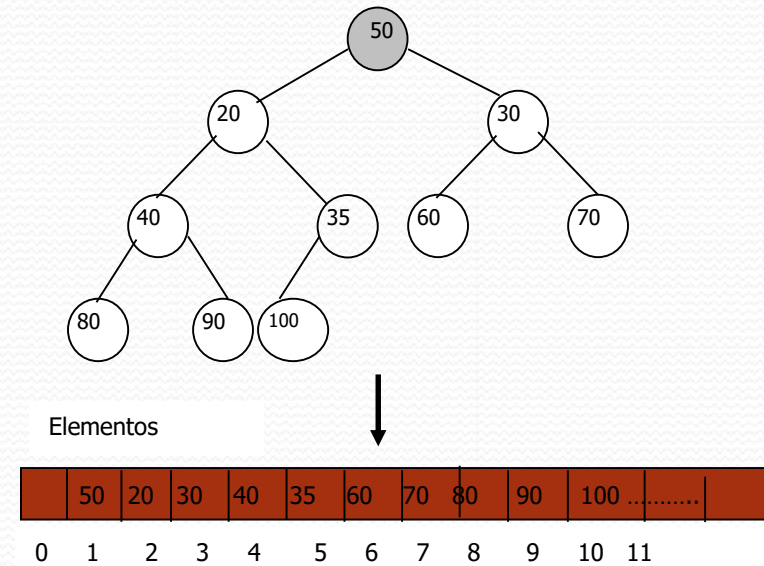
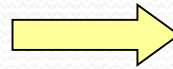
T.A.D. Montículo Binario

Construcción de operaciones abstractas (3)

Eliminar_Mínimo (M, X)



a) Propiedad de Estructura

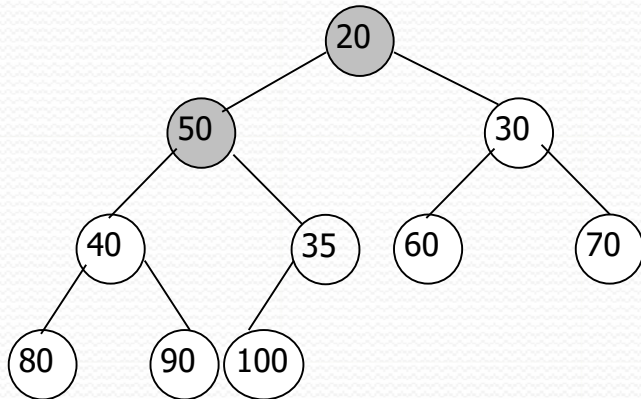


T.A.D. Montículo Binario

Construcción de operaciones abstractas (4)

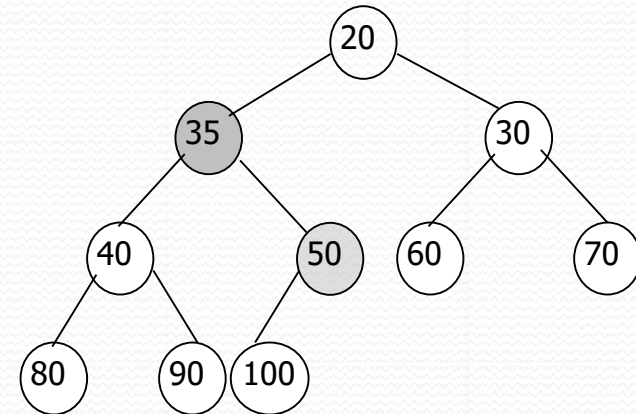
Eliminar_Mínimo (M, X)

b) Propiedad de Orden



Elementos

0	1	2	3	4	5	6	7	8	9	10	11
	20	50	30	40	35	60	70	80	90	100



Elementos

0	1	2	3	4	5	6	7	8	9	10	11
	20	35	30	40	50	60	70	80	90	100