



Algoritmos y Estructura de Datos

Unidad 2 – Semana 10



Logro de sesión

Al finalizar la sesión, el estudiante el estudiante diseña árboles binarios para el almacenamiento y recuperación de datos tomando en cuenta el tiempo de acceso.



Sesión 6 : Árboles binarios

Contenido:

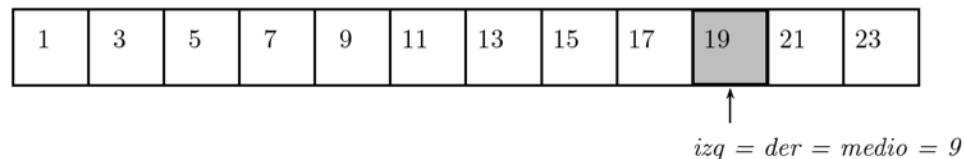
- Búsqueda binaria
- Árboles binarios



Búsqueda binaria



- El arreglo inicial:*



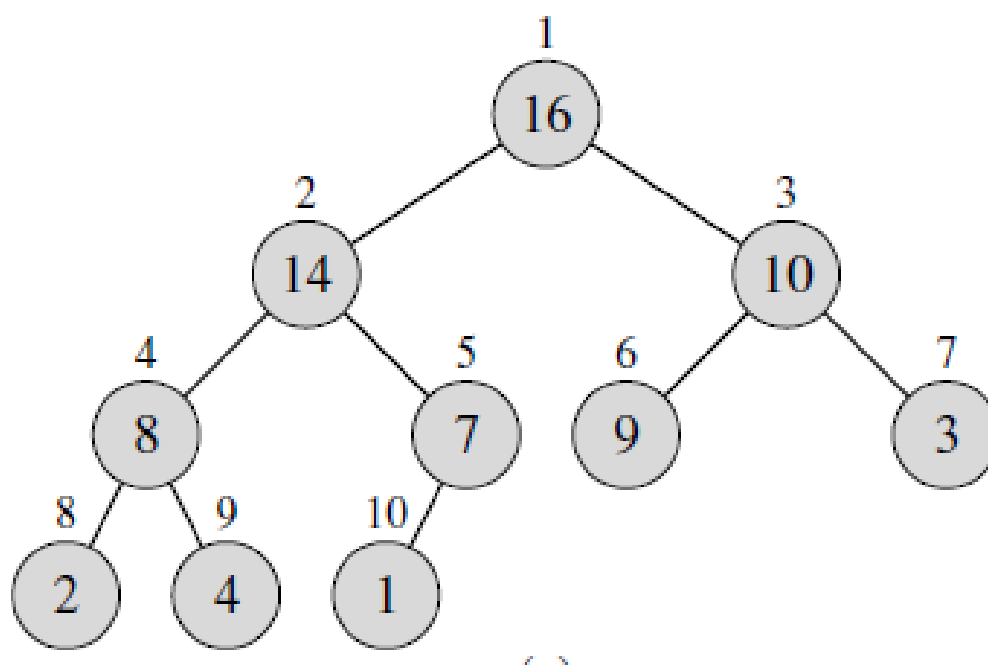
Búsqueda binaria



```
1  /**
2   * Performs the standard binary search using two comparisons per level.
3   * Returns index where item is found or -1 if not found.
4   */
5  template <typename Comparable>
6  int binarySearch( const vector<Comparable> & a, const Comparable & x )
7  {
8      int low = 0, high = a.size( ) - 1;
9
10     while( low <= high )
11     {
12         int mid = ( low + high ) / 2;
13
14         if( a[ mid ] < x )
15             low = mid + 1;
16         else if( a[ mid ] > x )
17             high = mid - 1;
18         else
19             return mid;    // Found
20     }
21     return NOT_FOUND;    // NOT_FOUND is defined as -1
22 }
```



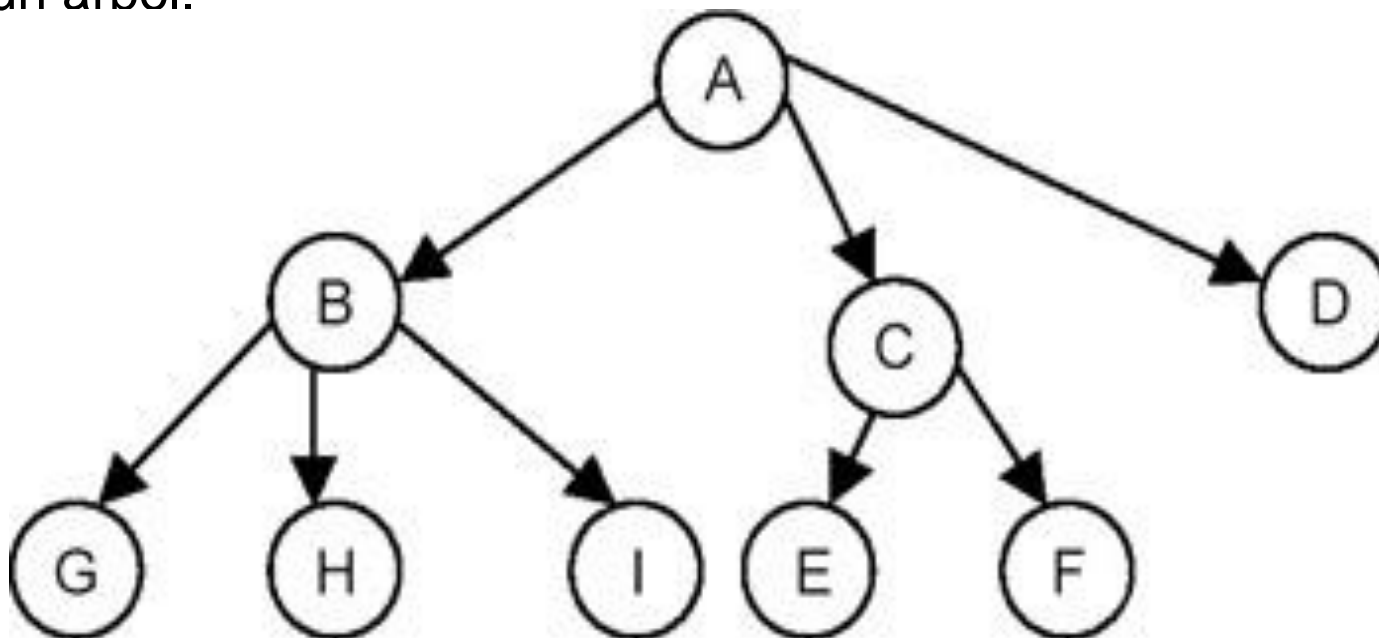
Arboles



Árbol



- Es una estructura de datos muy usada porque denota jerarquía.
- Estructura de Datos muy utilizada para búsquedas y ordenamientos, en diferentes aplicaciones (bases de datos, redes, etc.).
- Un árbol debe cumplir estas condiciones:
 - Siempre hay un Root Node o Nodo Raíz : el primero en el árbol.
 - Los demás Nodos forman conjuntos disjuntos los cuales a su vez pueden ser un árbol.



Árbol



- **Child Node o Nodo Hijo:** nodo debajo de o contenido en otro.
- **Parent Node o Nodo Padre:** nodo sobre de o que contiene otro.
- Se denomina **Leaf Node o Nodo Hoja** a aquellos que no tienen hijos (los últimos del árbol).
- Si cada nodo puede tener n hijos, siendo denominado según el número de máximo que puede tener como n -ario: unitario, binario, ternario, etc.
- Una lista enlazada vista de manera horizontal se podría interpretar como un árbol unitario.

Árbol



- Se denomina **subárbol** a la colección de nodos cuya raíz es uno de los nodos hijo.
- La línea que conecta un nodo padre con su hijo es llamada **RAMA**.
- El **TAMAÑO** de un nodo, es el numero de todos sus descendientes incluyéndose así mismo
- El **GRADO** de un nodo es el numero de hijos que tiene un nodo.
- Los nodo hoja tienen grado 0.

Árbol Binario

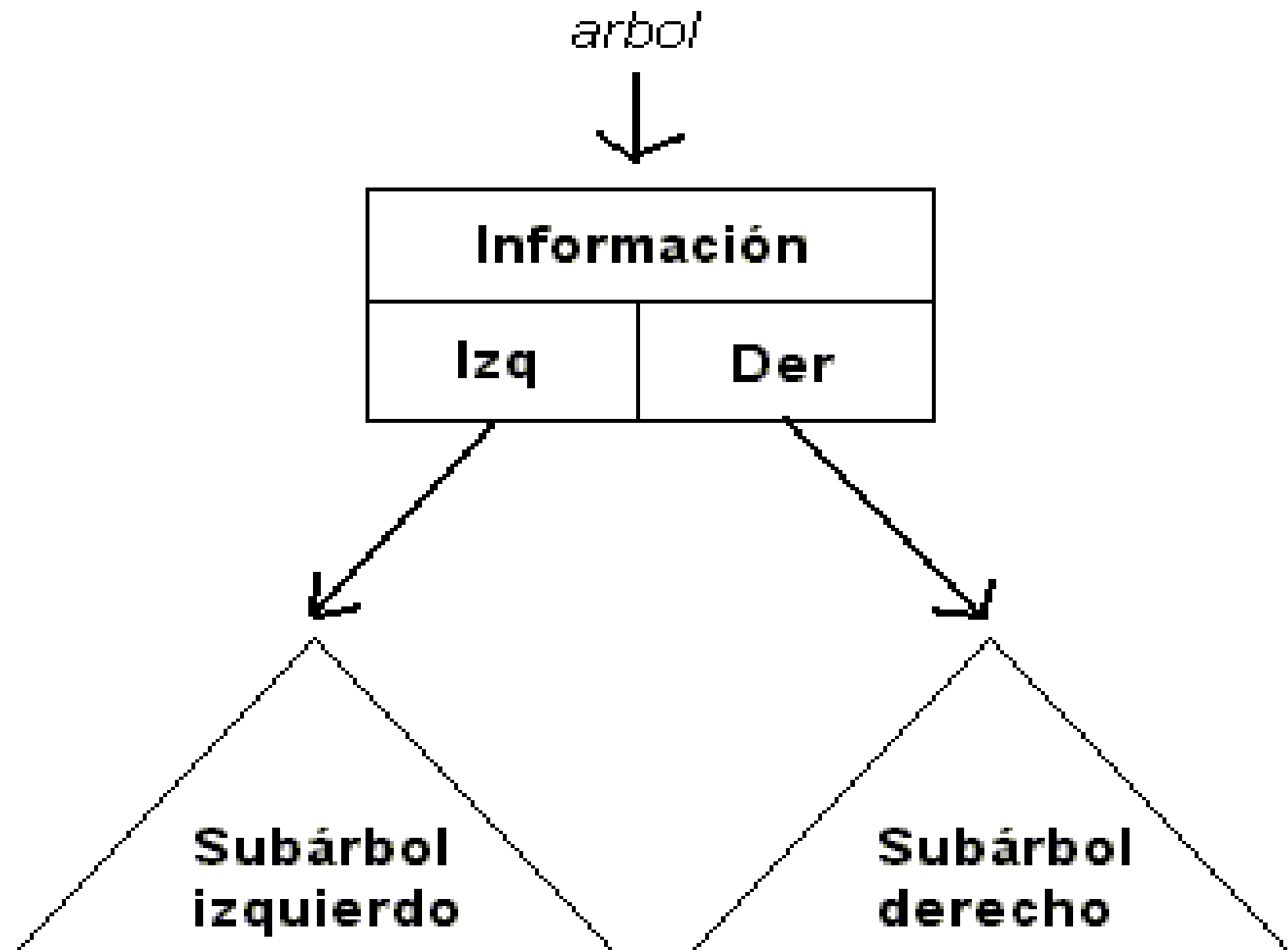


- Es un árbol cuya propiedad es que cada nodo puede tener hasta dos hijos. El grado de cada nodo es como máximo 2.
- Por convención, los nodos hijos se denominan nodos hijo izquierdo y derecho.
- El número máximo de nodos en un nivel i es: 2^i
- Si k es la profundidad del árbol, lo máximo que un árbol puede tener de hijos es:

$$2^k - 1 = 2^{k-1} + 2^{k-2} + \dots + 2^0$$

Fuente [P.S. Deshpande and O.G. Kakde. C and Data Structure. 2004]

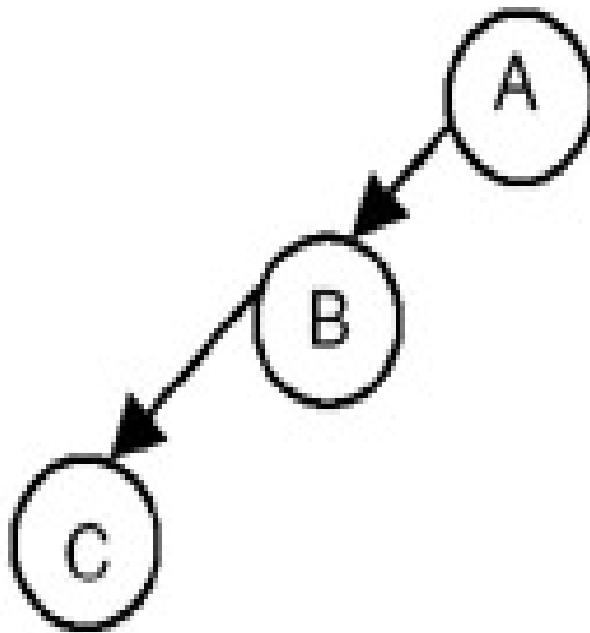
Árbol Binario



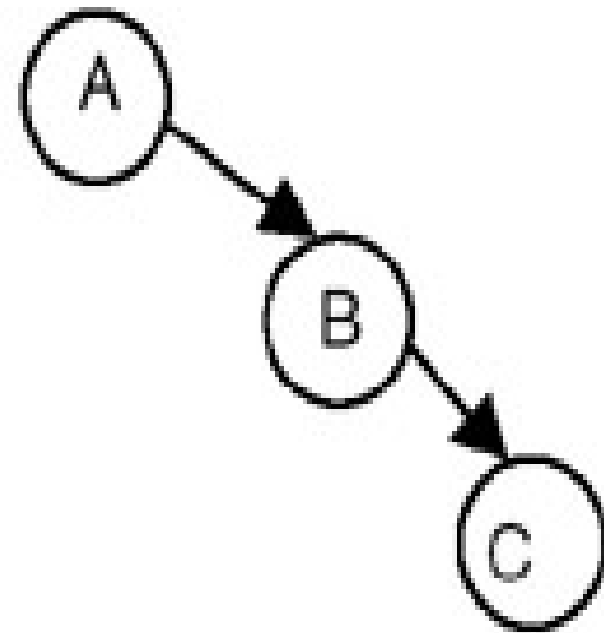
Árbol Binario



- Otros ejemplos



Left Skewed



Right Skewed

Árbol Binario



ALTURA

- La ALTURA de un nodo es el número de ramas que existen entre el nodo y la hoja mas “lejana” de esta.
- Todas las hojas tienen altura 0.
- La altura de un árbol es la altura de la raíz.
- Si T es un árbol binario, Altura(T) es:

Si T.vacia entonces

result \leftarrow 0

Caso contrario

result \leftarrow 1 + max(Altura(T.Izquierdo),
Altura(T.Derecho))

Árbol Binario



Nivel (Profundidad)

- El nivel de un nodo X esta definido como, numero de ramas para alcanzar al padre:

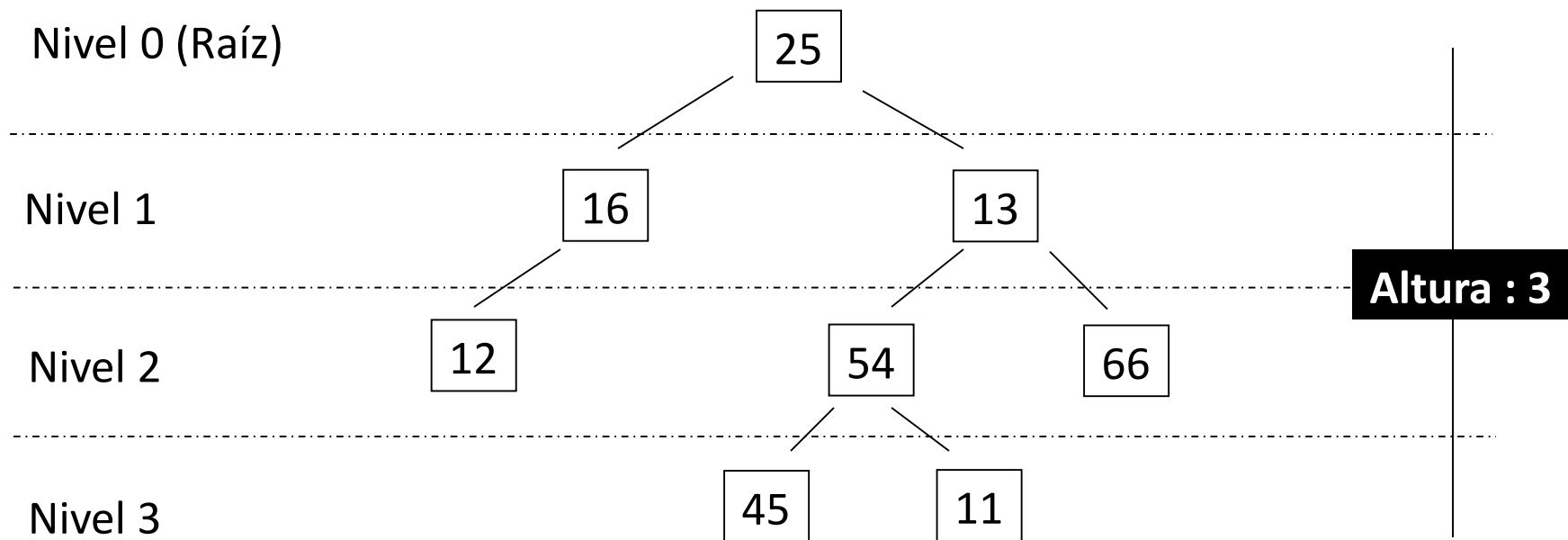
Si $X.EsRaiz$ entonces

$result \leftarrow 0$

Caso contrario

$result \leftarrow 1 + Nivel(X.Padre)$

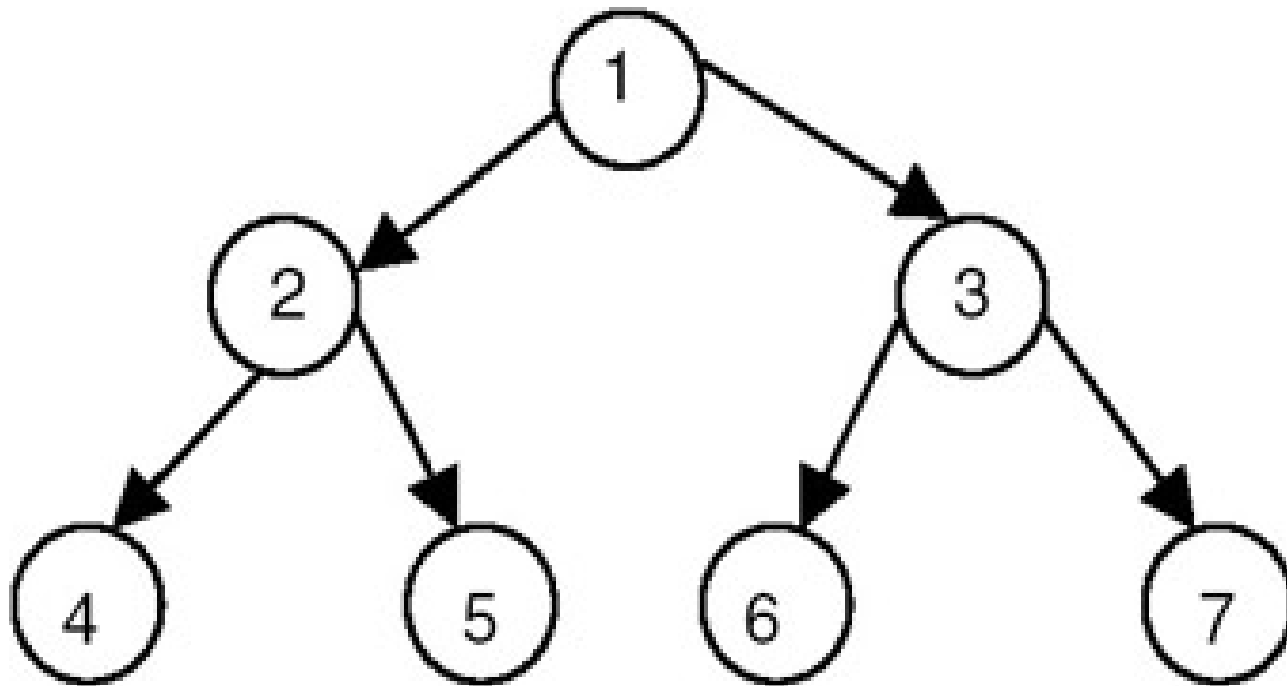
Árbol Binario



Árbol Binario



Árbol binario lleno



Si $k=3$, luego el número de nodos $\text{nodes} = 7$

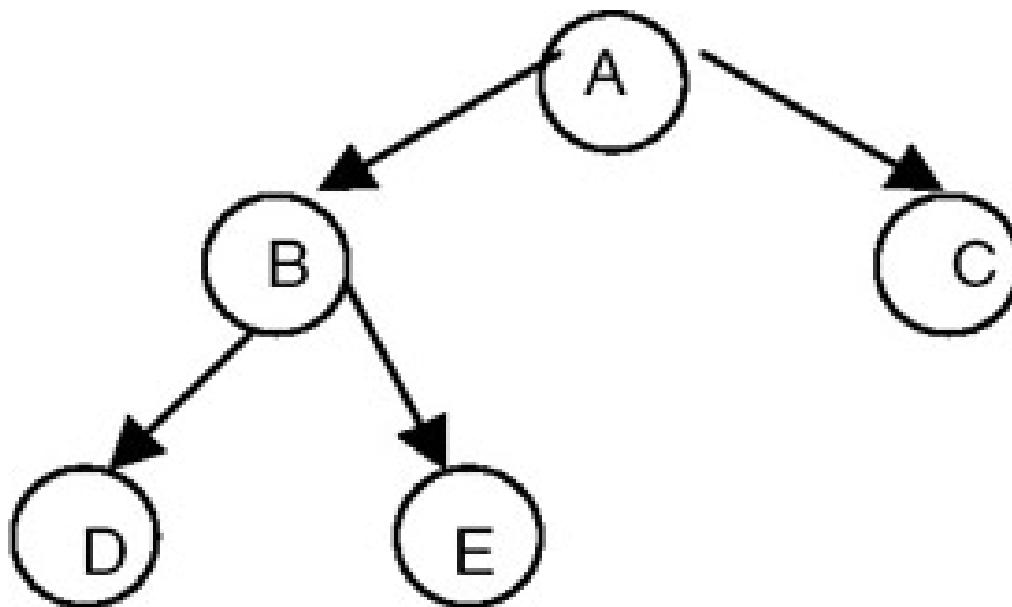
$$2^k - 1$$

K es la altura de la raíz + 1

Árbol Binario



Árbol binario completo



Todos los nodos hoja están al mismo nivel n o al menos en $n-1$. Siempre los hijos izquierdo están completos, el de la derecha puede no estarlo.

Altura de C?

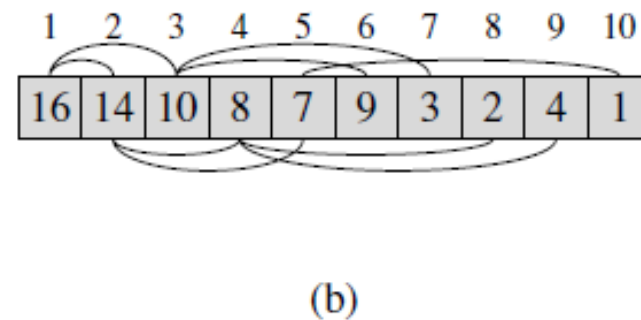
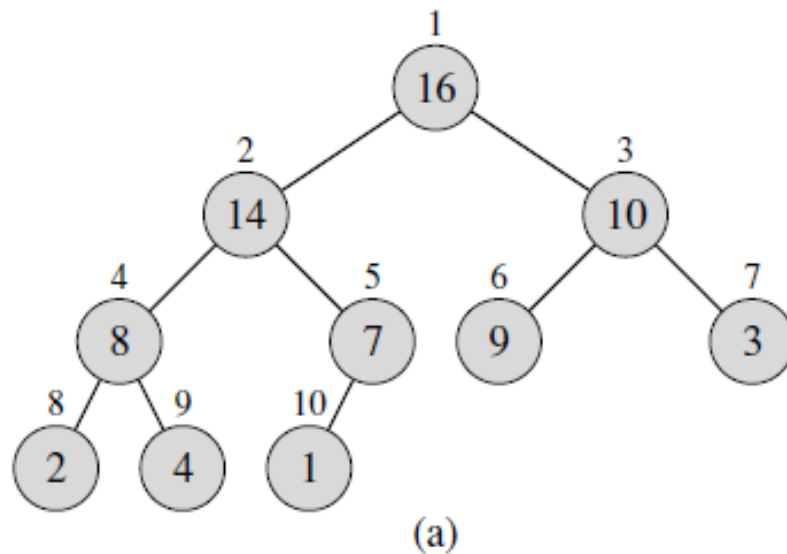
Altura de B?

Profundidad de C?

Árbol Binario



- Representación:



- (a) Un árbol binario
- (b) Su arreglo

Nodo i debe almacenarse en posición i

El padre de un nodo i , está en el índice $i/2$

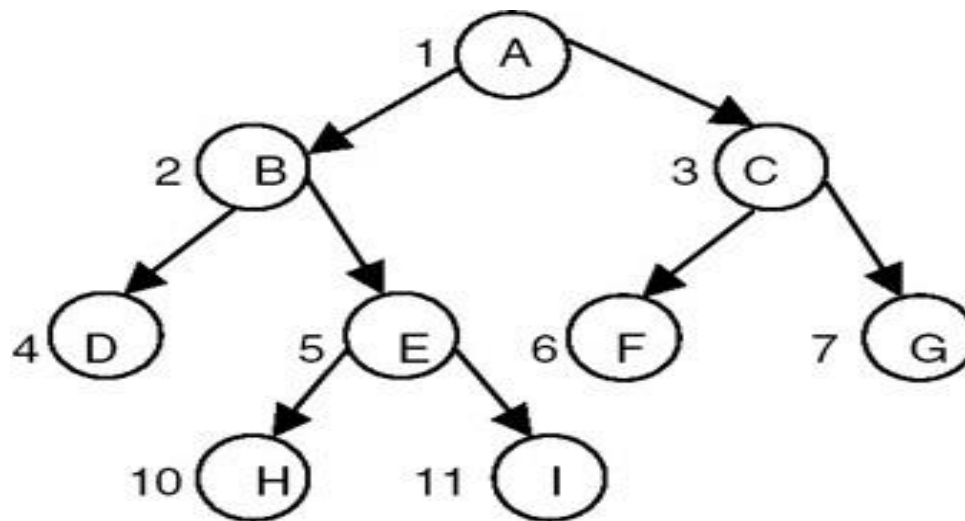
El hijo izquierdo está en la posición $2i$

El hijo derecho está en la posición $2i + 1$

Árbol Binario



Arreglo para Árbol incompleto



1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	
9	
10	H
11	I

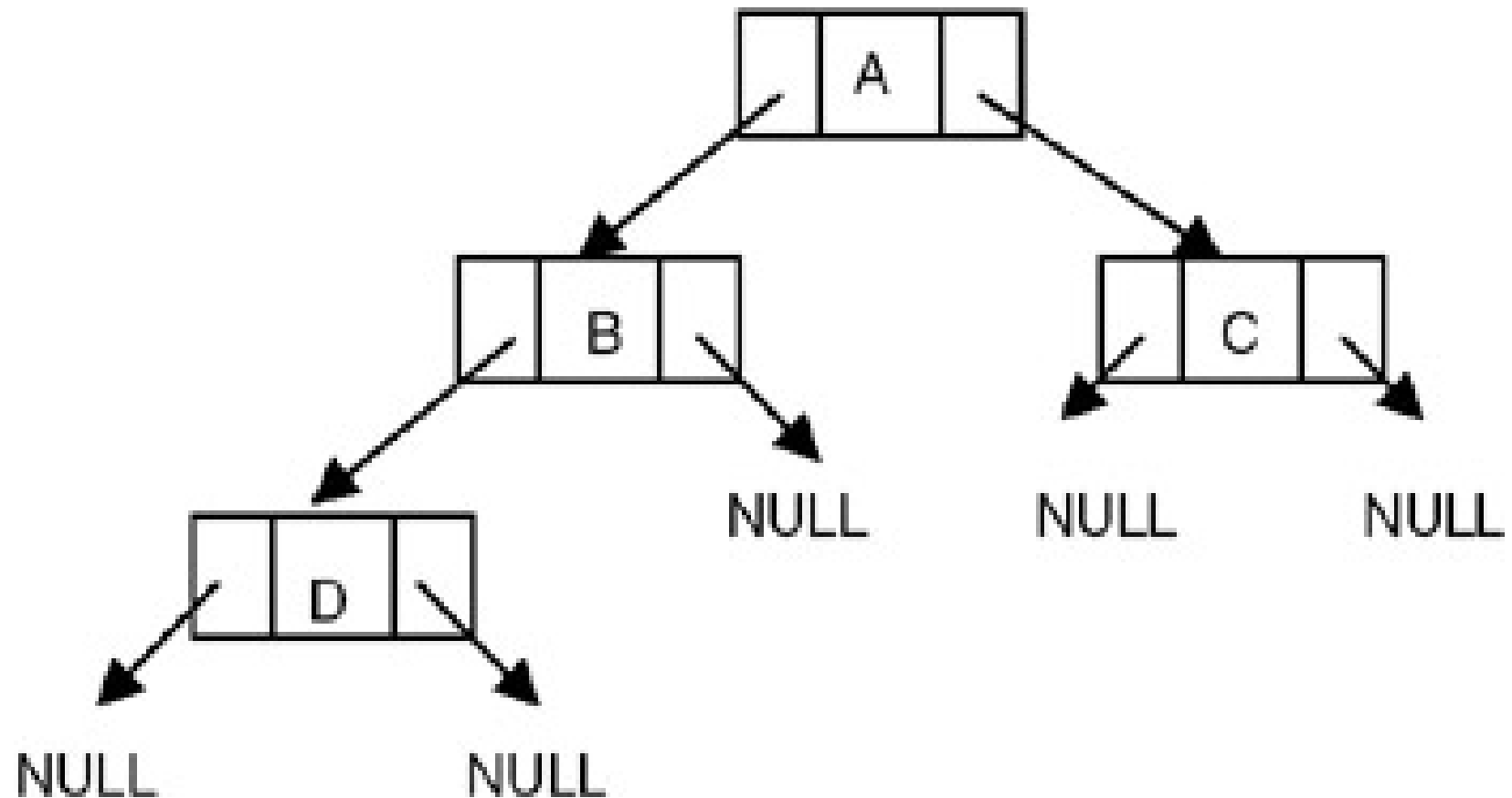
Array tree

Se pierde espacio

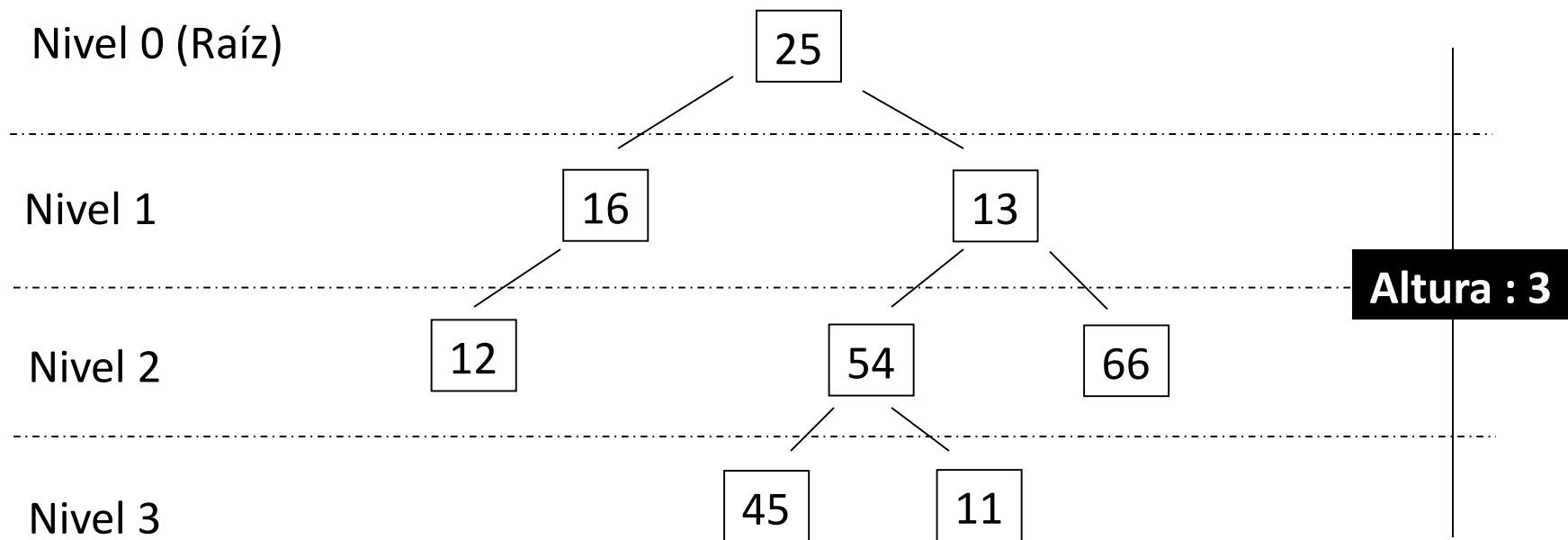
Árbol Binario



Representando Árbol con listas



Árbol Binario



Árbol Binario



Recorrido de un árbol

- PreOrden
 - Visita la raíz, luego recorre en PreOrden el nodo a la izquierda y finalmente el de la derecha.
 - Ej : 25, 16, 12, 13, 54, 45, 11, 66
 - Una búsqueda que usa un recorrido PreOrden es llamado “depth-first” o “profundidad-primero”.
- InOrden
 - Recorre en InOrden primero el subárbol de la izquierda, luego procesa la raíz, y finalmente el subárbol de la derecha.
 - Ej : 12, 16, 25, 45, 54, 11, 13, 66
- PostOrden
 - Recorre en PostOrden primero el subárbol de la izquierda, el subárbol de la derecha y finalmente la raíz.
 - Ej : 12, 16, 45, 11, 54, 66, 13, 25

Árbol Binario



Recorrido de un árbol

- Por Nivel
 - Cada nivel del árbol es recorrido en orden : primero la raíz, luego todos los nodos de izquierda a derecha en el nivel 1, y así sucesivamente hasta que se recorran todos los nodos.
 - Es también llamado “breadth-first” o “ancho-primero”.
 - Sencillo de conceptualizar pero difícil de codificar. Requiere el uso de una pila.
 - Ej : 25, 16, 13, 12, 54, 66, 45, 11

Referencias



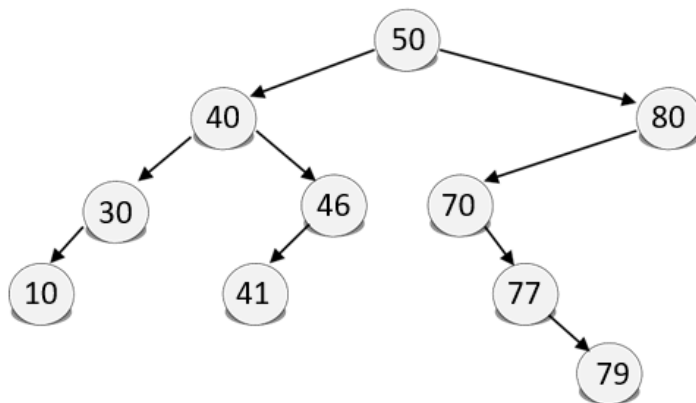
- ❑ Sedgewick, R., et. al. (2011) Algorithms, Fourth Edition. Pearson.
- ❑ Cormen, H., et. al. (2009) Introduction to Algorithms, MIT Press.
- ❑ Allen, Mark (2014) Data Structures and Algorithms Analysis in C++, Fourth Edition. Pearson.

Ejercicios

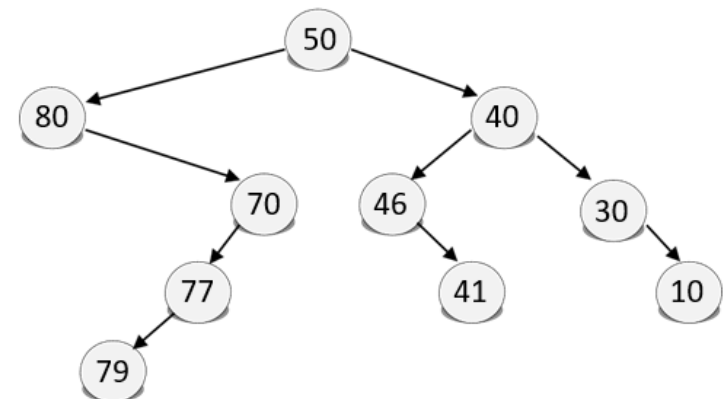


- Implemente una función que determine cuantos nodos hay al lado izquierdo y cuantos hay al lado derecho de la raíz.
- Implemente una función que devuelva la suma de todos los números de un árbol binario de enteros.
- Implemente una función que permita convertir un árbol binario en su espejo.

ANTES



DESPUES





EXIGETE INNOVA