



Universidad Autónoma del Estado de México
UAEM



CENTRO UNIVERSITARIO
UAEM ZUMPANGO

INGENIERÍA EN COMPUTACIÓN

Unidad de Aprendizaje:

ESTRUCTURA DE DATOS



Unidad de Competencia III:

ESTRUCTURA NO LINEAL - ÁRBOLES

M. en C. Edith Cristina Herrera Luna



ÁRBOLES BINARIOS

ESTRUCTURA DE DATOS

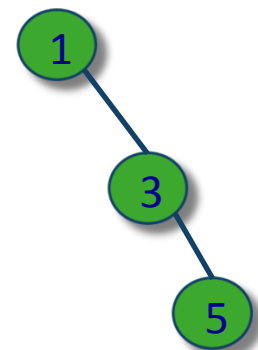
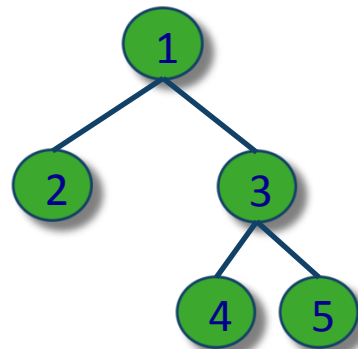
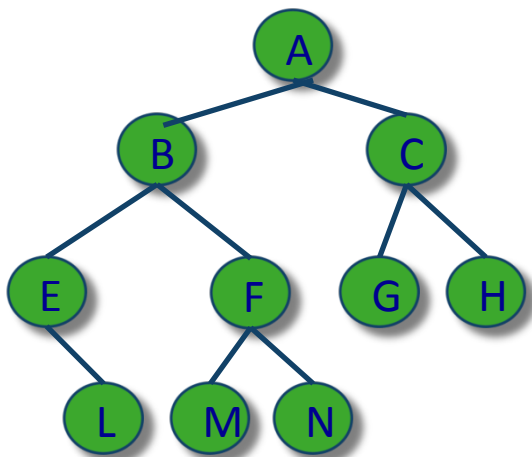
ÁRBOL BINARIO

+ Es un árbol que tiene como máximo dos nodos sucesores (nodos hijos) en cada nodo del árbol.

+ Una definición formal es:

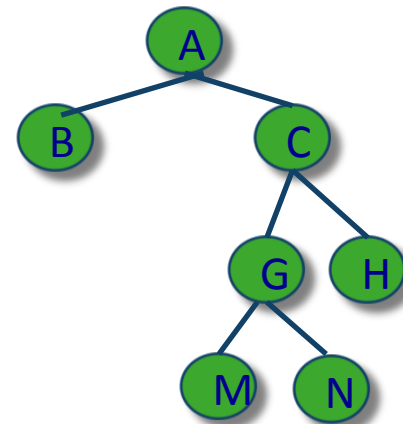
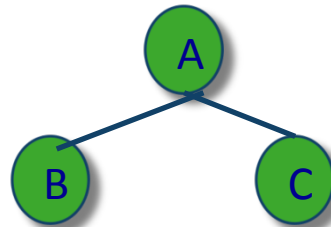
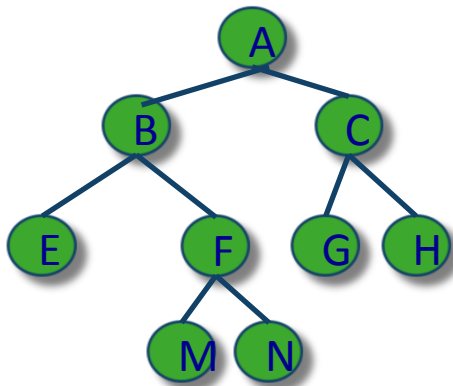
Un conjunto de nodos T es un árbol binario si cualquiera de las siguientes condiciones se cumple:

- T está vacío
- Si T no está vacío, >ene un nodo raíz r con 0, 1 ó 2 subárboles binarios cuyas raíces se encuentran conectadas a r por una rama.

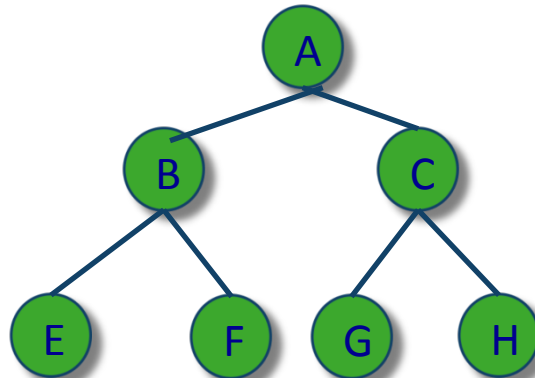


ÁRBOL BINARIO LLENO Y PERFECTO

- + ARBOL BINARIO LLENO: Es un árbol binario en el que todos sus nodos internos tienen exactamente dos hijos.

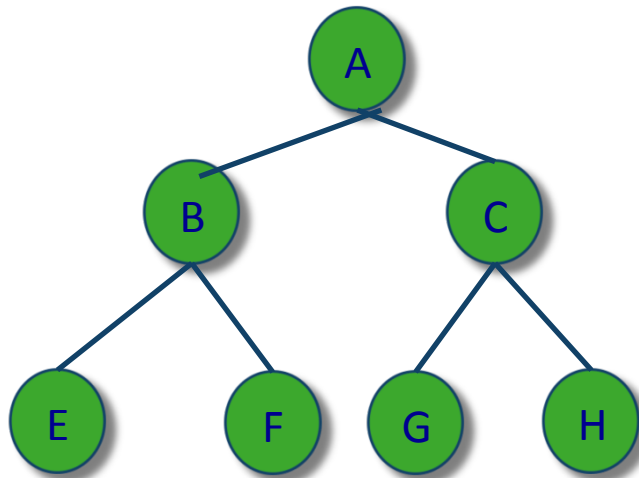


- + ARBOL BINARIO PERFECTO: Es un árbol lleno con todas sus hojas en el mismo nivel.



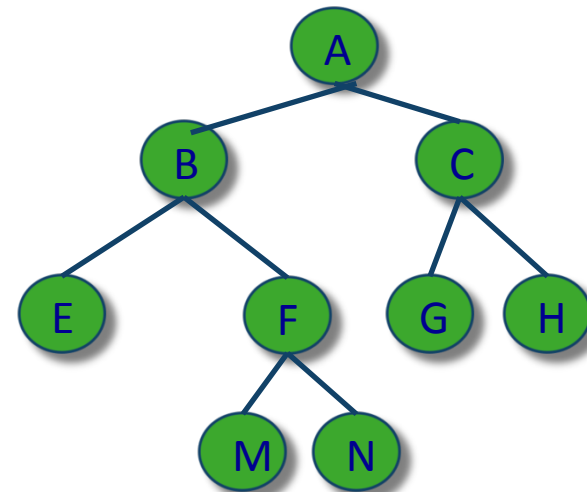
ÁRBOL BINARIO LLENO Y PERFECTO

- + Un árbol binario lleno tiene $2m-1$ nodos, donde m representa el número de nodos hoja en el árbol.



$$m = 4$$

$$2(4) - 1 = 7$$

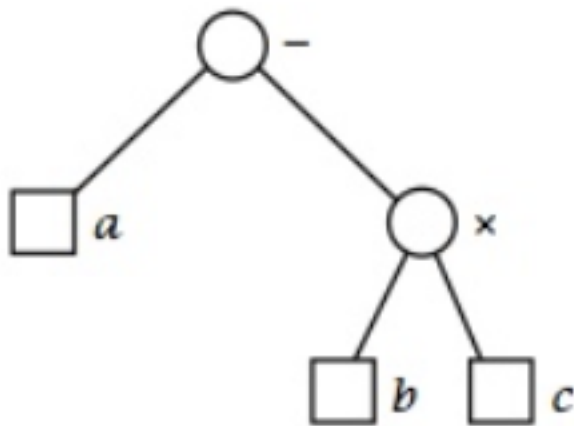


$$m = 5$$

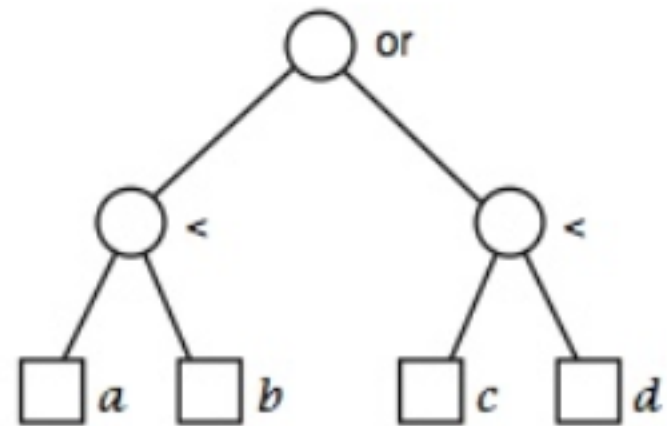
$$2(5) - 1 = 9$$

ÁRBOLES DE EXPRESIÓN

- + Son árboles binarios que representan expresiones, nos ayudan a identificar la jerarquía de operaciones. Generalmente los nodos terminales representan operandos, mientras que los nodos internos son operadores.



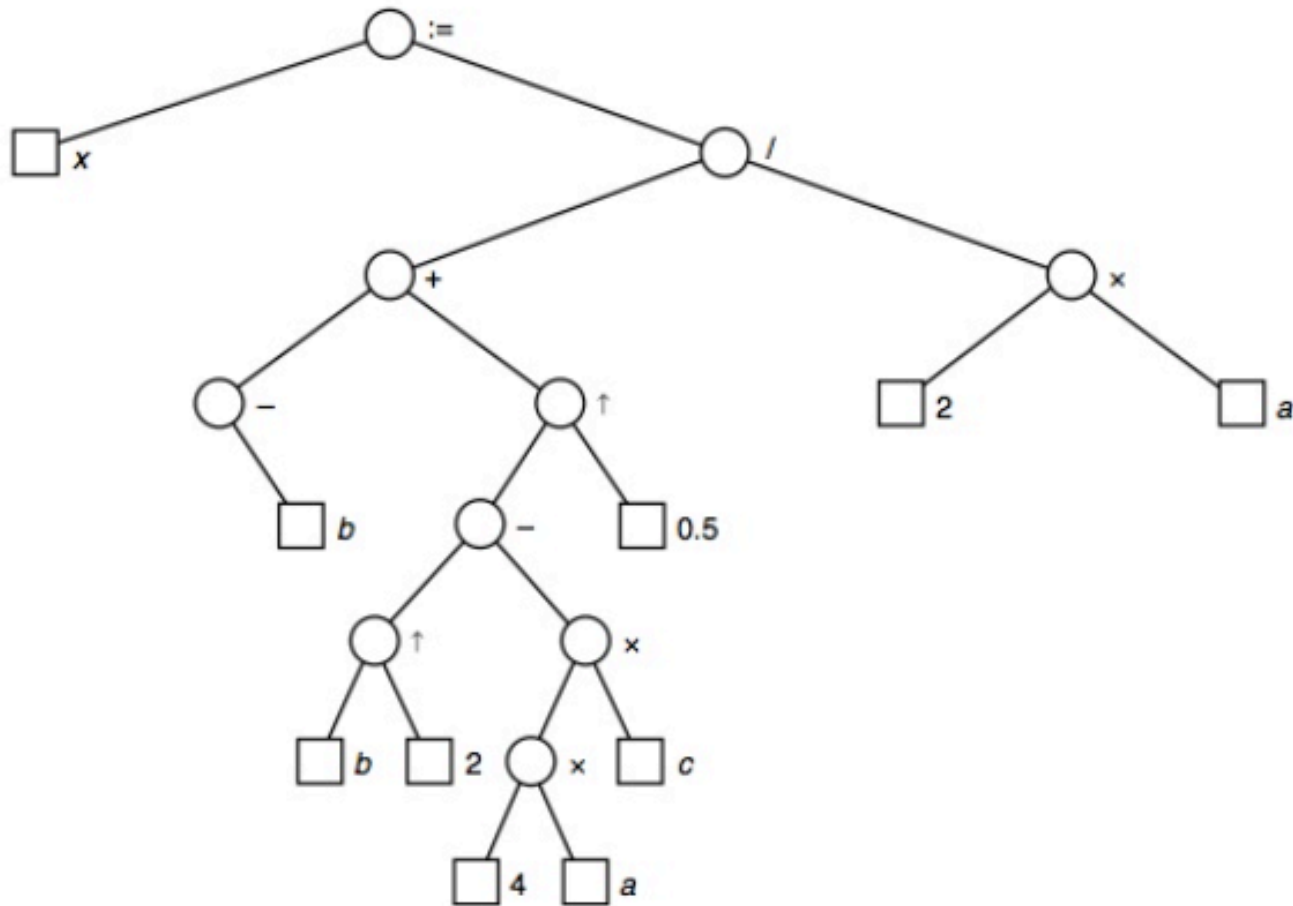
$a - (b \times c)$



$(a < b) \text{ or } (c < d)$

ÁRBOLES DE EXPRESIÓN

+ ¿Que expresión representa el siguiente árbol?



ÁRBOLES DE EXPRESIÓN

Dibuja un árbol de expresión para:

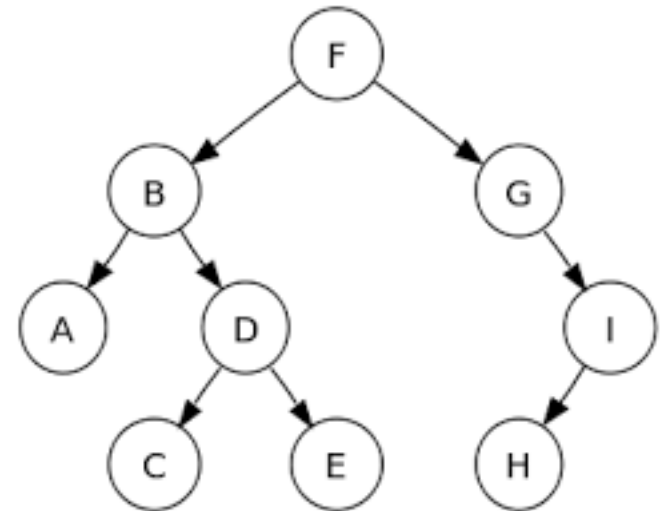
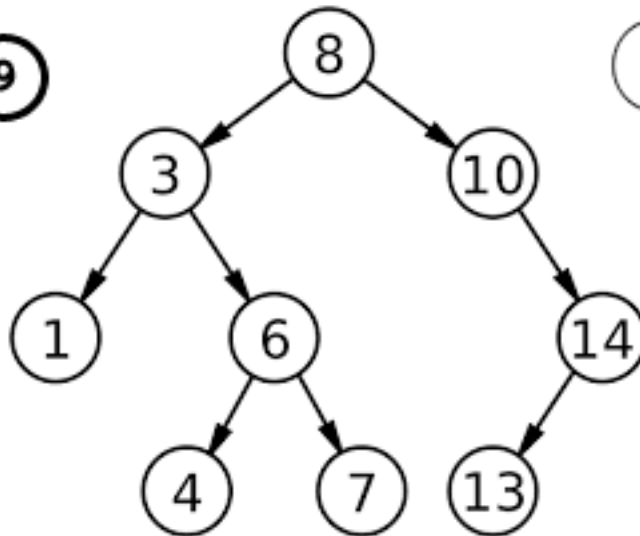
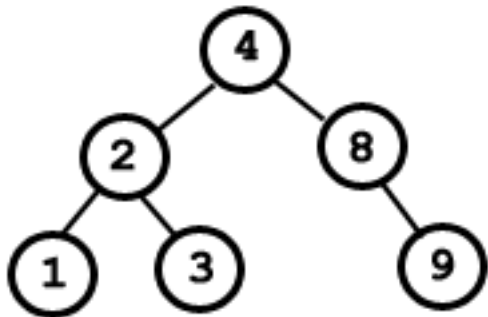
+ $(x + y) * (x / z) + n$

+ $i * (j + k) + (j * m) / (n - k)$

+ $(a < b) \&\& (b < c) \|\| (c < (5 + x))$

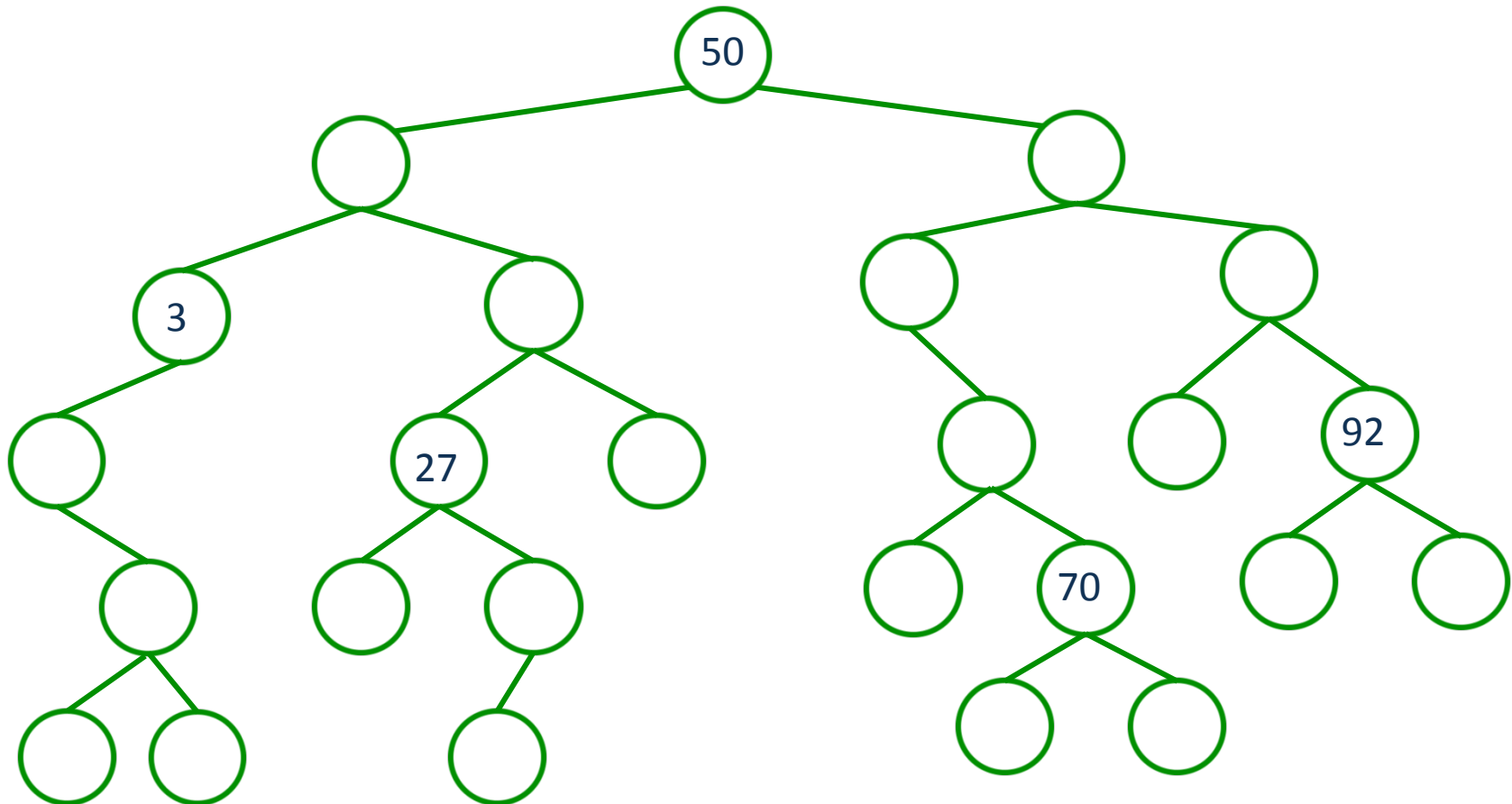
ÁRBOLES DE BÚSQUEDA

- + Es un árbol binario donde cada nodo tiene en su subárbol izquierdo nodos con claves menores y en su subárbol derecho nodos con claves mayores a él.



ÁRBOLES DE BÚSQUEDA

+ Agrega los elementos faltantes al siguiente árbol de búsqueda

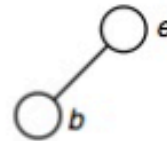


ÁRBOLES DE BÚSQUEDA

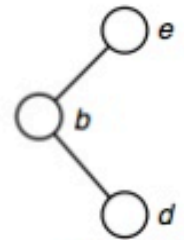
+ Para insertar elementos en un árbol de búsqueda, se debe comparar el elemento a insertar con el nodo raíz, de acuerdo a si es mayor o menor se sigue un subárbol y se inserta el nuevo elemento hasta encontrar un nodo con cero o un hijo.



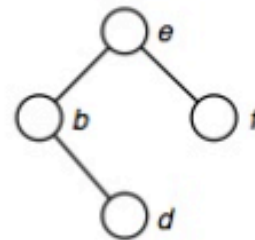
(a) Insert *e*



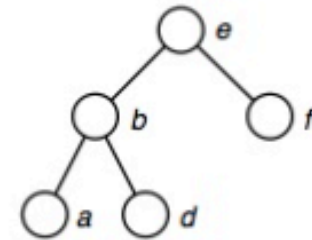
(b) Insert *b*



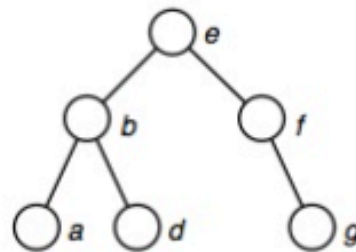
(c) Insert *d*



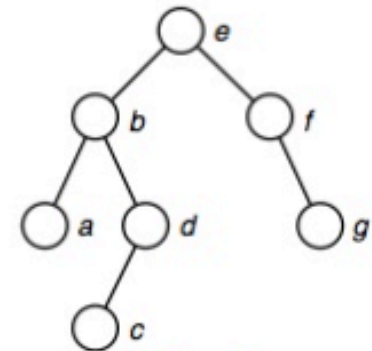
(d) Insert *f*



(e) Insert *a*



(f) Insert *g*



(g) Insert *c*



RECORRIDOS EN ÁRBOLES BINARIOS

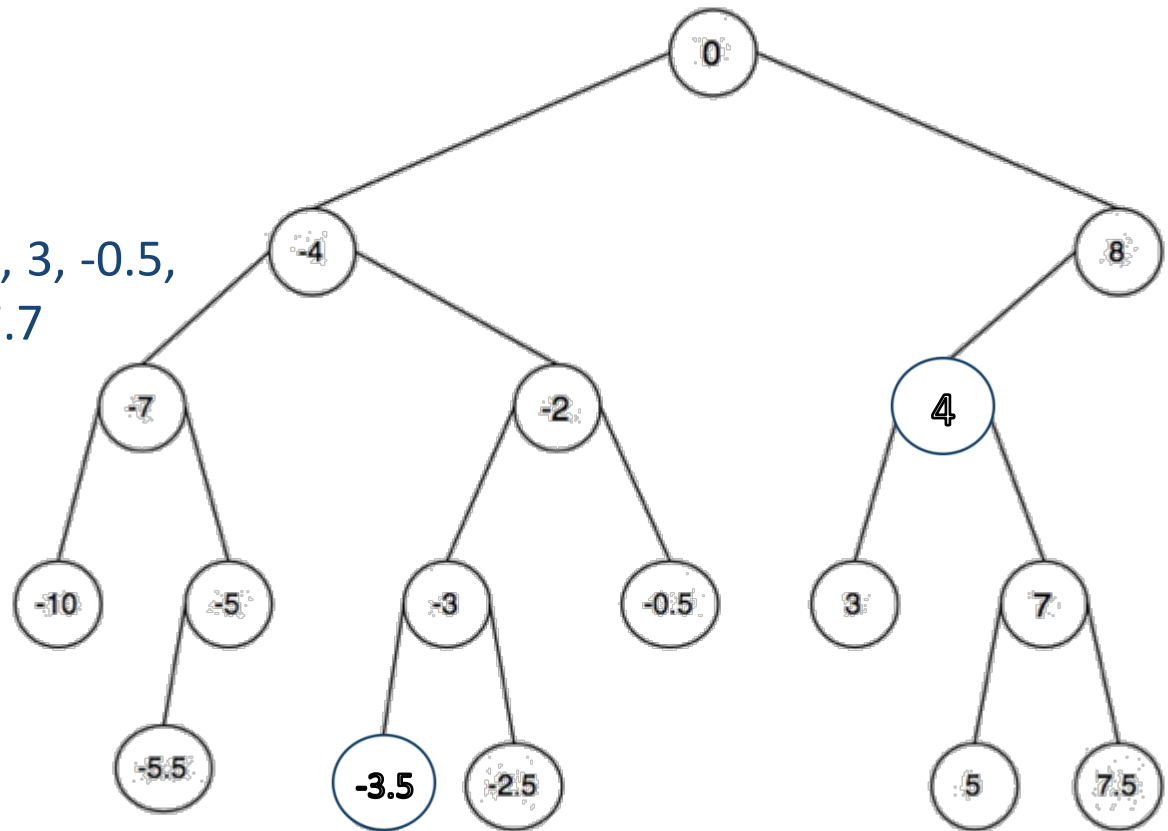
ESTRUCTURA DE DATOS

RECORRIDOS EN ÁRBOLES BINARIOS

+ POR AMPLITUD O POR NIVEL:

Se recorre de izquierda a derecha los nodos, desde el nivel superior hasta el ultimo nivel.

0, -4, 8, -7, -2, 4, -10, -5, 3, -0.5,
3, 7, -5.5, -3.5, -2.5, 5, 7.7

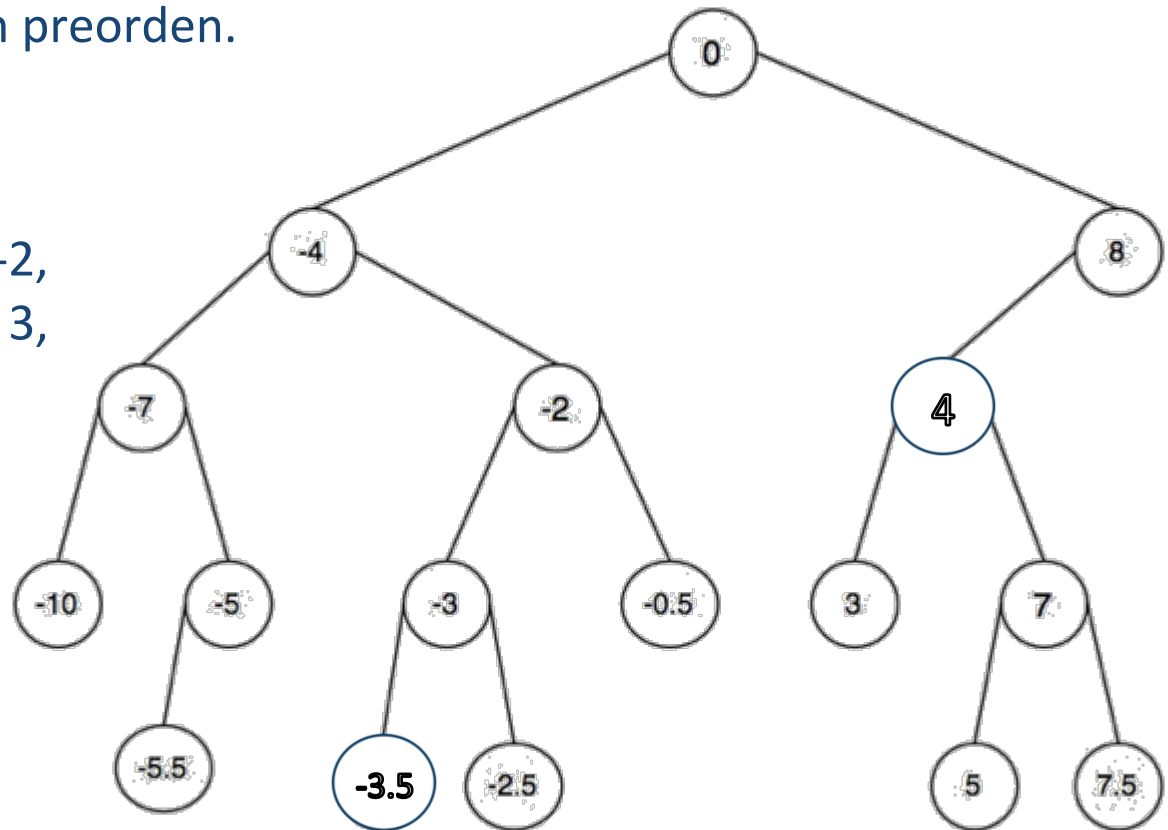


RECORRIDOS EN ÁRBOLES BINARIOS

+ PREORDEN: R - I - D

Se recorre la raíz, el subárbol izquierdo en preorden y finalmente el subárbol derecho en preorden.

0, -4, -7, -10, -5, -5.5, -2,
-3, -3.5, -2.5, -0.5, 8, 4, 3,
7, 5, 7.5.



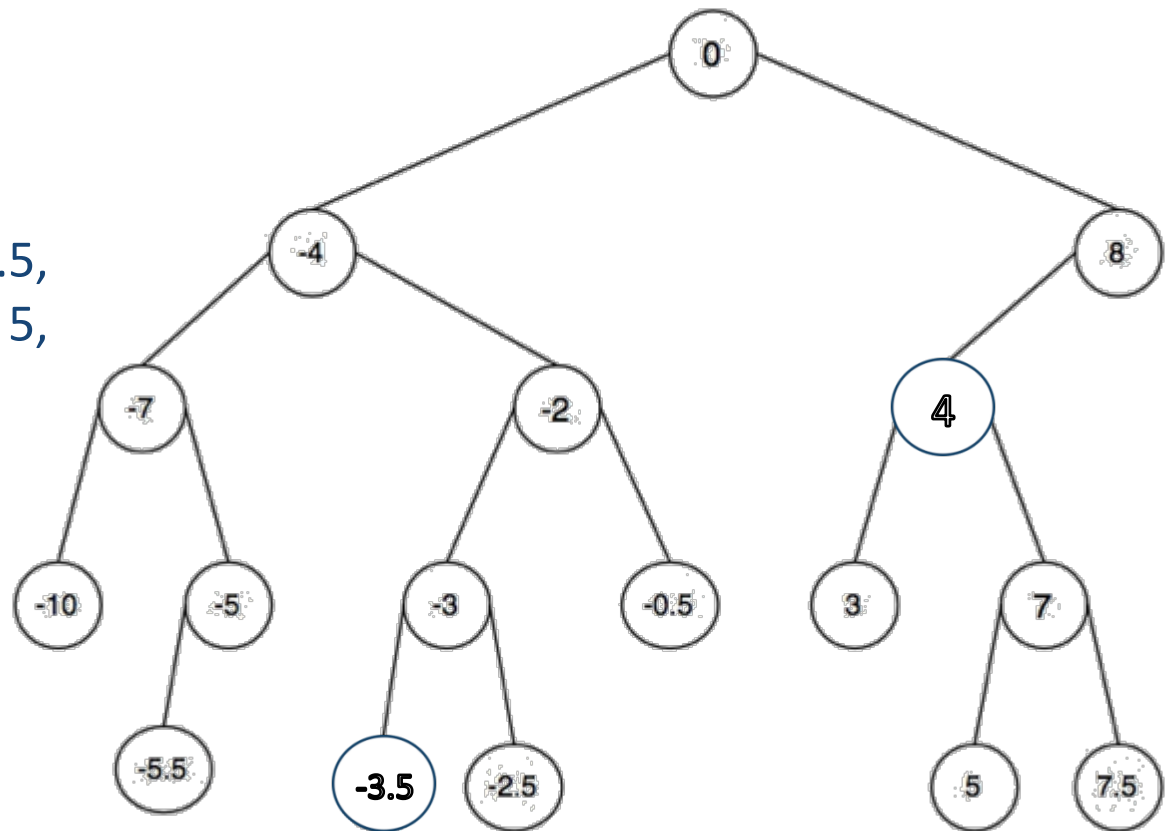
RECORRIDOS EN ÁRBOLES BINARIOS

+ INORDEN:

I – R – D

Se recorre el subárbol izquierdo en inorden, la raíz y el subárbol derecho en inorden.

-10, -7, -5.5, -5, -4, -3.5,
-3, -2.5, -2, -0.5, 0, 3, 4, 5,
7, 7.5, 8.



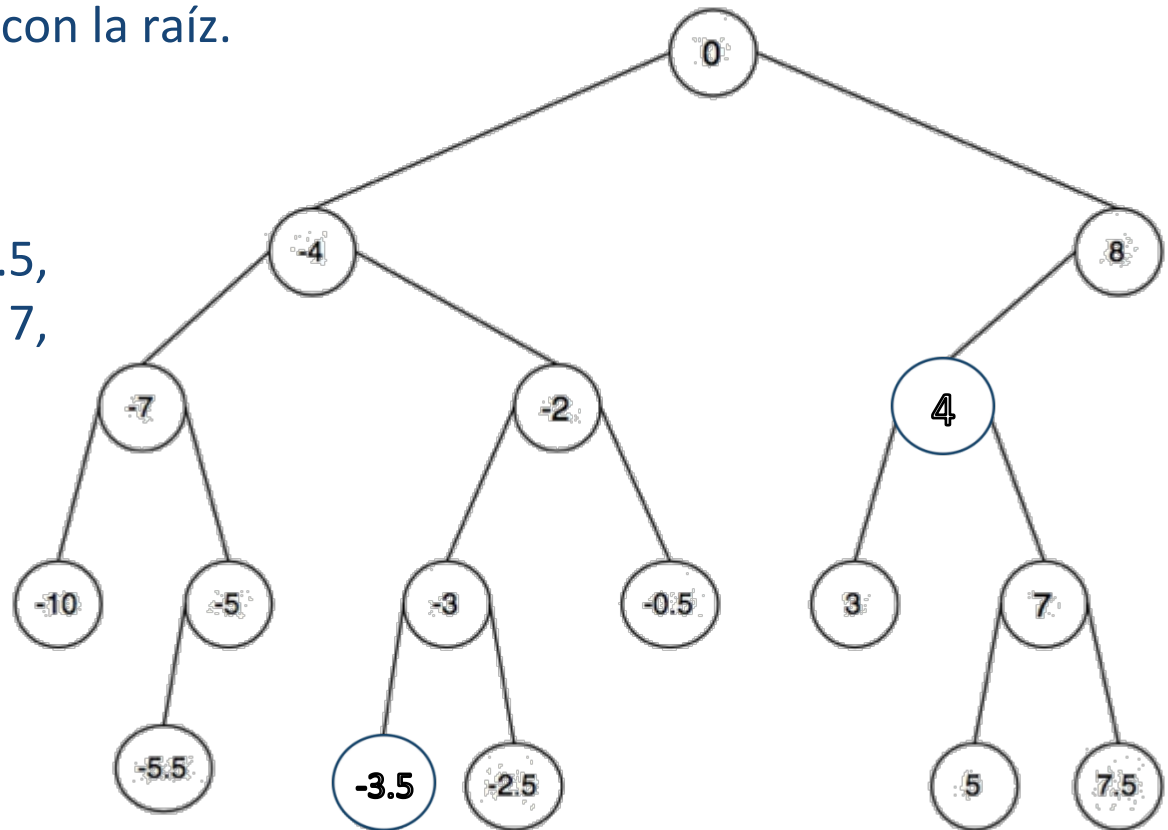
RECORRIDOS EN ÁRBOLES BINARIOS

+ POSTORDEN:

I - D - R

Se recorre el subárbol izquierdo en postorden, el subárbol derecho en postorden y finaliza con la raíz.

-10, -5.5, -5, -7, -3.5, -2.5,
-3, -0.5, -2, -4, 3, 5, 7.5, 7,
4, 8, 0.



RECORRIDOS EN ÁRBOLES BINARIOS

+ Preorden: $R - I - D$

+ Inorden: $I - R - D$

+ Postorden: $I - D - R$

Existen los mismos recorrido en orden **converso**, recorriendo de derecha a izquierda:

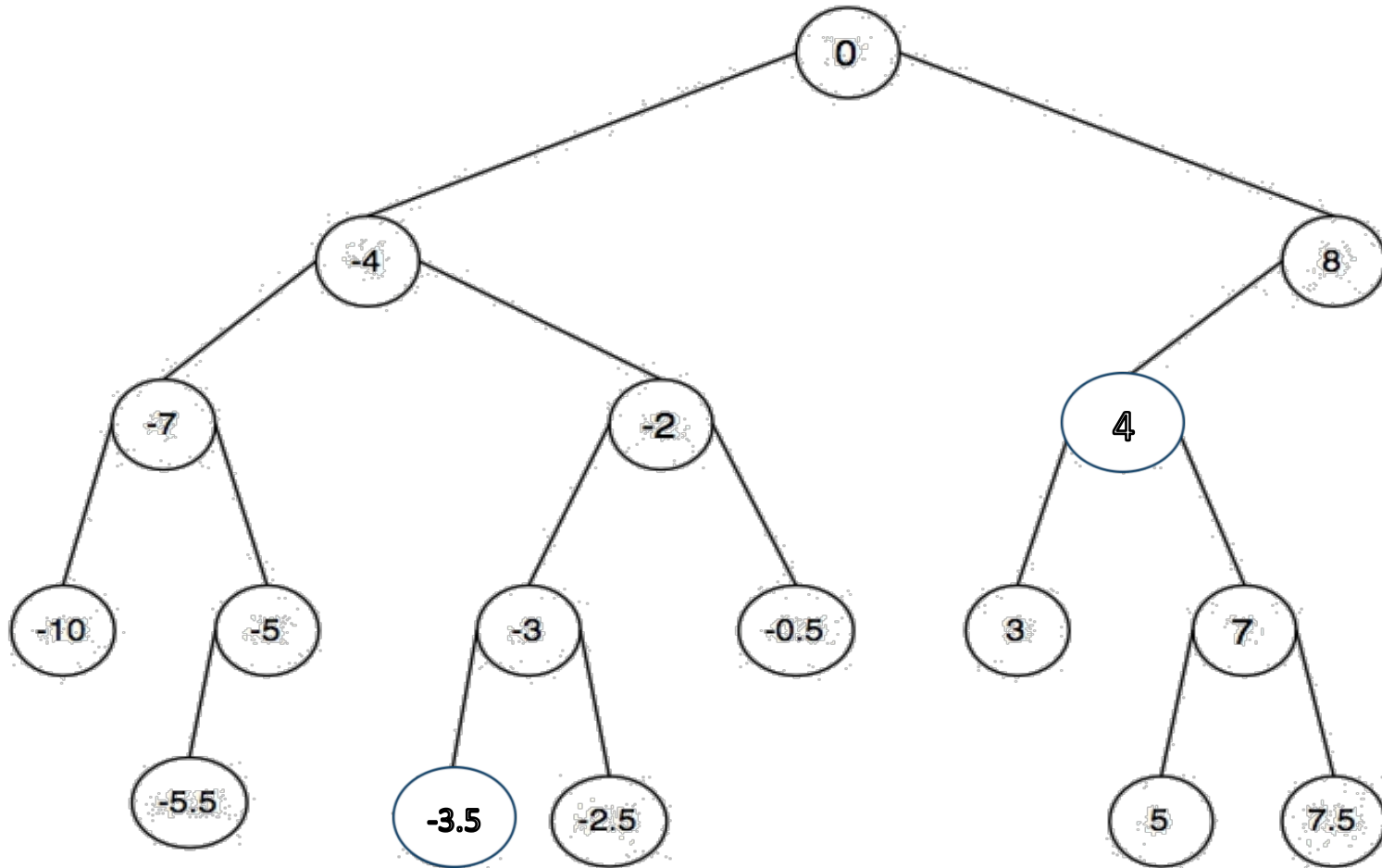
+ Preorden converso: $R - D - I$

+ Inorden converso: $D - R - I$

+ Postorden converso: $D - I - R$

RECORRIDOS EN ÁRBOLES BINARIOS

+ Escribe los recorridos conversos para el siguiente árbol



RECORRIDOS EN ÁRBOLES BINARIOS

+ Preorden converso:

0, 8, 4, 7, 7.5, 5, 3, -4, -2, -0.5, -3, -2.5, -3.5, -7, -5, -5.5, -10

+ Inorden converso:

8, 7.5, 7, 5, 4, 3, 0, 0.5, -2, -2.5, -3, -3.5, -4 -5, -5.5, -7, -10

+ Postorden converso:

7.5, 5, 7, 3, 4, 8, -0.5, -2.5, -3.5, -3, -2, -5.5, -5, -10, -7, -4, 0



REFERENCIAS

PROGRAMACIÓN 1

REFERENCIAS

- + Koffman, Elliot y Wolfgang, Paul. (2008). Estructura de datos con C++. Objetos, abstracciones y diseño. McGraw-Hill.
- + Cairó, Osvaldo y Guardati, Silvia. (2006). Estructuras de datos (3a. Edición). McGraw-Hill.
- + Criado, Ma. Asunción. (2006). Programación en lenguajes estructurados. AlfaOmega Ra-Ma.
- + Joyanes, Luis. (2008). Fundamentos de programación (4a Edición). McGraw-Hill.
- + López, Leobardo. (2004). Programación estructurada. Un enfoque algorítmico (2a. Edición). AlfaOmega.
- + Joyanes, Luis. M. Fernández, L: Sánchez, I. Zahonero. (2005). Estructuras de datos en C. McGraw-Hill. Schaum.
- + Web:C Plus Plus: Biblioteca STL www.cplusplus.com/reference/stl .
- + Savitch, Walter. Absolute Java. Pearson. ISBN-13: 978-0134041674



GRACIAS

ESTRUCTURA DE DATOS
Continua Unidad de Competencia IV: GRAFOS

GUÍA PARA EL PROFESOR

- + Las primeras diapositivas muestran el propósito, justificación y objetivos de la unidad de aprendizaje. Se presentan para que el alumno identifique dichos elementos.
- + El contenido, conforme a la unidad de aprendizaje, maneja los temas de un menor a mayor grado de dificultad.
- + Se recomienda que el alumno realice pruebas de escritorio para los ejercicios de recursión. Es conveniente verificar qué pasa por hacer cambios simples como las impresiones antes y después de las llamadas recursivas.
- + Para los ejercicios de recursión se sugiere iniciar con funciones con tipo de retorno void y posteriormente tratar diversos tipos de datos.
- + Para los árboles y su programación se recomienda la plantilla STL de C++ Standard Template Library, posteriormente pedir que se programen usando arreglos y al final pedir que en base a estructuras (struct) el alumno defina sus propias estructuras.
- + Los ejercicios de recorrido sobre arboles deben realizarse primero diseñando el árbol, incorporando valores al árbol de manera ordenada y posteriormente hacer los recorridos. Realizar ejercicios de arboles con diversos tipos de datos para los nodos: números, letras, palabras, etc.