

# Árboles Binarios de Búsqueda (*Binary Search Trees*)

Programación de Estructuras de Datos y Algoritmos Fundamentales  
(TC1031)

M.C. Xavier Sánchez Díaz  
sax@tec.mx



# Outline

## 1 Árboles

- Elementos de un árbol

## 2 Árboles binarios de búsqueda (Binary Search Trees)

# ¿Qué es un árbol?

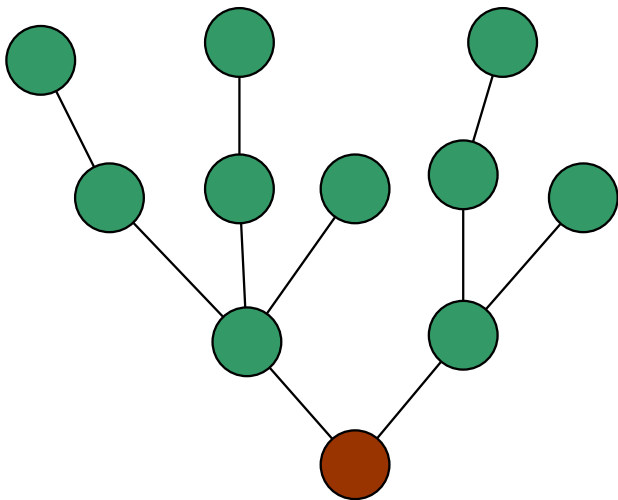
## Árboles

Un **árbol** es una **estructura de datos** **jerárquica** y **recursiva**:

- **Jerárquica** porque tiene *niveles* o *prioridades*
- **Recursiva** porque puede describirse recursivamente: *el hijo del hijo del hijo...*



# Árboles

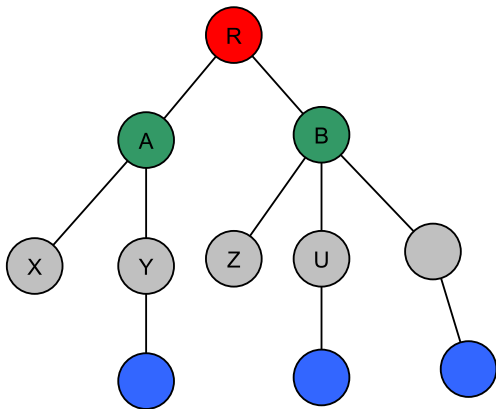




# Raíz

## Elementos de un árbol

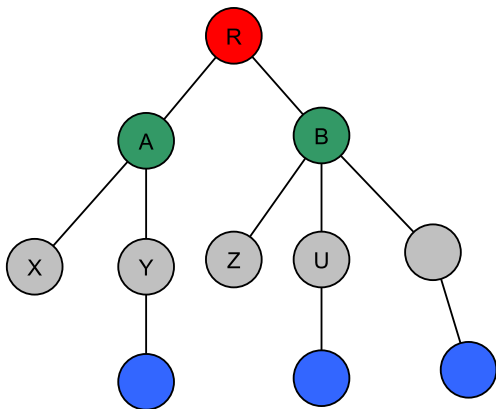
La **raíz** es el *inicio* del árbol; el **nodo** que es padre de todos.



# Nodos y relaciones

## Elementos de un árbol

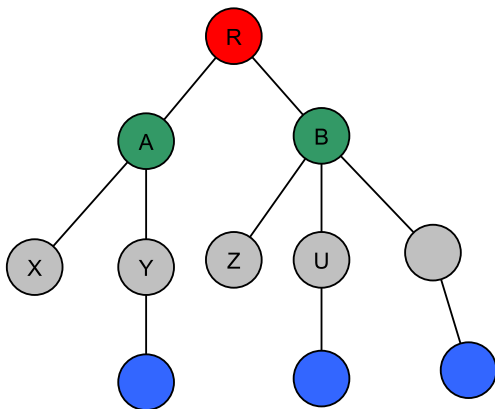
El elemento  $R$  es **padre** (*parent*) de los **nodos**  $A$  y  $B$ .  $A$  y  $B$  son **hijos** (*children*) de  $R$ .  $A$  y  $B$  son hermanos (*siblings*).



# Niveles

## Elementos de un árbol

*A* y *B* están en el mismo nivel. *X*, *Y*, *Z* y *U* están en el mismo nivel.

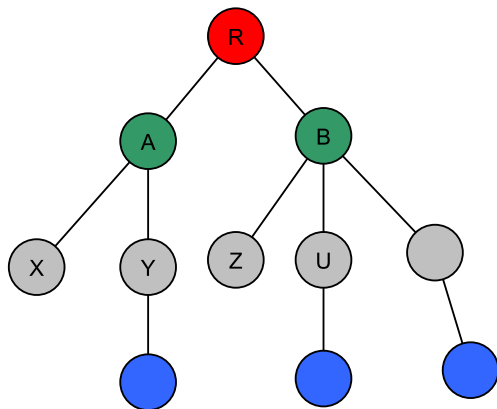




# Jerarquía

## Elementos de un árbol

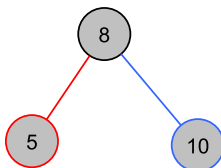
Los elementos **sin descendencia** son los nodos **hoja**.



# Árboles binarios de búsqueda

Un **árbol binario de búsqueda** (*binary search tree*) es un árbol que es:

- **Binario** porque cada nodo tiene a lo mucho **2** hijos.
- **de Búsqueda** porque como si fuera *binary search*, los descendientes **izquierdos** deben ser **más chicos** que el padre, y los descendientes **derechos** deben ser **más grandes** que el padre.



Teniendo un árbol binario de  $n$  nodos donde cada uno (a excepción de las hojas) tuvieran dos hijos cada quién, ¿cuál sería la altura del árbol? es decir, ¿Cuántos niveles tendría?

# Operaciones posibles

## Árboles binarios de búsqueda

Las sencillitas:

- Buscar (un valor)
- Agregar (un valor)
- Borrar (un valor)

Las complicadas:

- Obtener la altura (del árbol)
- Obtener los ancestros (de un dato)
- Obtener el nivel (de un dato)

Las *extremadamente* complicadas<sup>1</sup>:

- Recorridos

---

<sup>1</sup>no es cierto

# Operaciones posibles

## Árboles binarios de búsqueda

Las sencillitas:

- Buscar (un valor)
- Agregar (un valor)
- Borrar (un valor)

Las complicadas:

- Obtener la altura (del árbol)
- Obtener los ancestros (de un dato)
- Obtener el nivel (de un dato)

Las *extremadamente* complicadas<sup>1</sup>:

- Recorridos

---

<sup>1</sup>no es cierto

# Operaciones posibles

## Árboles binarios de búsqueda

Las sencillitas:

- Buscar (un valor)
- Agregar (un valor)
- Borrar (un valor)

Las complicadas:

- Obtener la altura (del árbol)
- Obtener los ancestros (de un dato)
- Obtener el nivel (de un dato)

Las *extremadamente* complicadas<sup>1</sup>:

- Recorridos

---

<sup>1</sup>no es cierto