

# Matemáticas discretas II

Facultad de Ingeniería. Universidad del Valle

Marzo 2018

# Contenido

## 1 Introducción a los árboles

## 2 Recorridos en árboles

## 3 Árboles de expansión

4) Algoritmos sobre árboles

$\left\{ \begin{array}{l} \text{Busqueda amplitud} \\ \text{Busqueda profundidad} \end{array} \right.$

# Contenido

**1** Introducción a los árboles

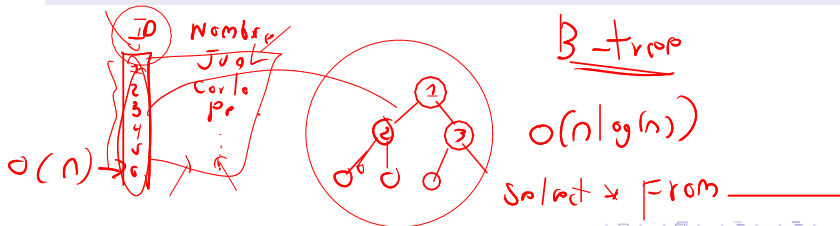
2 Recorridos en árboles

3 Árboles de expansión

# Introducción a los árboles

## Introducción

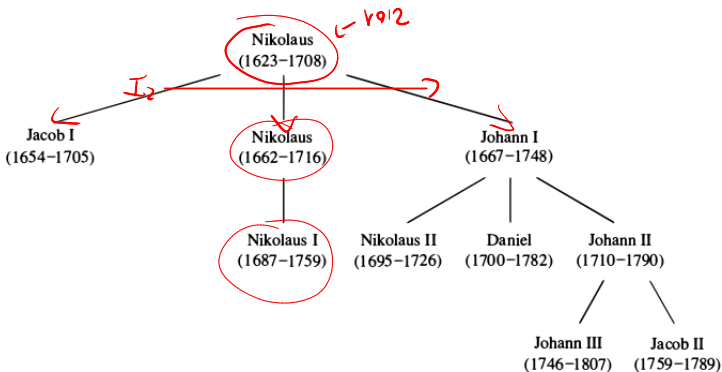
Un árbol es un grafo conexo que no contiene circuitos simples. Los árboles son utilizados en un gran número de problemas computacionales, como es el caso de algoritmos de codificación, programación dinámica, entre otros.



# Introducción a los árboles

## Definición 1

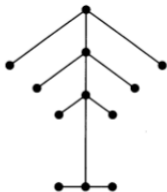
Un árbol no tiene ningún circuito simple. Esto quiere decir que no puede contener aristas múltiples ni ciclos.



# Introducción a los árboles

## Teorema 1

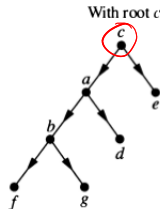
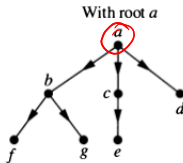
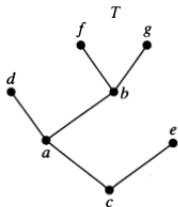
Un grado no dirigido es un árbol si y sólo si hay un único camino entre cada par de .



# Introducción a los árboles

## Definición 2

Un árbol con raíz es un árbol con un vértice que ha sido designado como raíz y cada arista se puede acceder desde un camino directo desde la raíz.



# Introducción a los árboles

## Definición 3

La terminología de los árboles tiene orígenes botánicos y genealógicos. Suponga que  $T$  es un árbol con raíz.

- El **padre** de  $v$  es el único vertice  $u$ , tal que hay una arista dirigida de  $u$  a  $v$
- El caso contrario anterior, se dice  $v$  es **hijo** de  $u$
- Los vértices con el mismo padre son llamados **hermanos**
- Los **antecesores** de cualquier vértices, son el camino desde la **raíz** hasta el vértices, pero excluyéndolo a él
- Los **descendientes** de un vértice son todos aquellos que tienen a  $v$  como antecesor



# Introducción a los árboles

## Definición 4

- Un vértice es llamado **hoja** si no tiene hijos
- Los vértices ~~de~~ <sup>que no tienen</sup> los hijos son llamados **vértices internos** / *nodos*

## Definición 5

El **nivel de un vértice** es la longitud del único camino desde la raíz que hasta él. El nivel de la raíz es 0

# Introducción a los árboles

## Definición 6

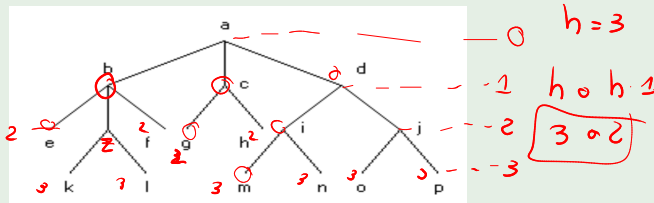
La **altura** de un árbol es la **longitud del camino más largo** desde la raíz hasta cualquier vértice

## Definición 7

Un árbol de altura ( $h$ ), está **equilibrado** o **balanceado** si todas sus hojas están en los niveles  $h$  o  $h - 1$

# Introducción a los árboles

## Ejemplo de árbol equilibrado

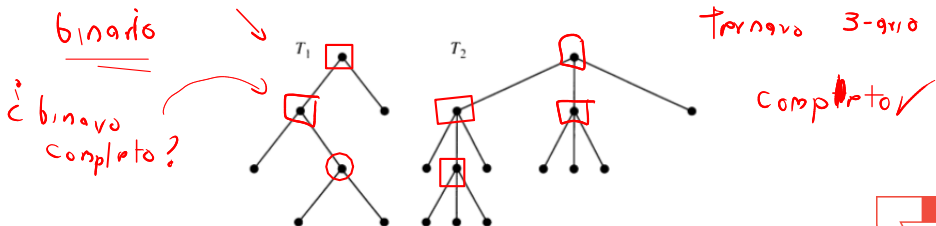


- 1 El árbol está **equilibrado**
- 2 Su **altura** es 3
- 3 j es el **padre** de p
- 4 Los **antecedentes** de n son  $\{i, d, a\}$
- 5 los **descendientes** de h son  $\{i, j, m, n, o, p\}$
- 6 Los vértices  $\{e, k, l, f, g, h, m, n, o, p\}$  son **hojas**
- 7 Los vértices  $\{a, b, c, d, i, j\}$  son **vértices internos**

# Introducción a los árboles

## Definición 8

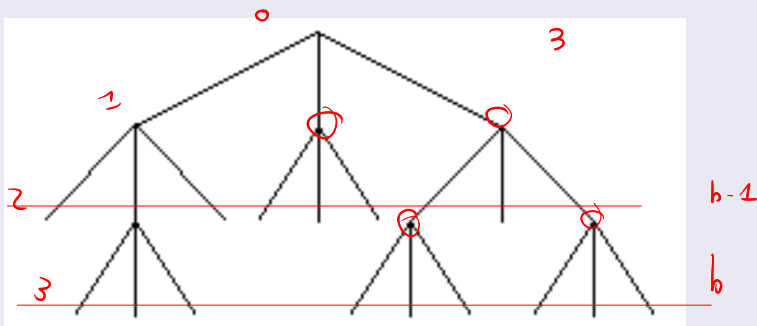
Un árbol con raíz es llamado  $m$ -ario si cada vértice interno no tiene más de  $m$  hijos. Un árbol es llamado un árbol  $m$ -ario completo si cada vértice interno tiene exactamente  $m$  hijos. Un árbol  $m$ -ario con  $m = 2$  es llamado árbol binario.



# Introducción a los árboles

## Definición 9

Un **árbol m-ario completo** es aquel donde los vértices internos tienen exactamente  $m$  hijos y es equilibrado.



# Introducción a los árboles



## Propiedades de los árboles

Un árbol con  $n$  vértices tiene  $n - 1$  aristas.

## Demostración

**Paso base:** Con  $n = 1$  se tiene 0 aristas.



**Paso inductivo:** Si se supone que para  $n$  existen  $n - 1$  aristas, ahora miramos para  $n + 1$ , para conectar el nuevo vértice se necesita una nueva arista

# Introducción a los árboles

## Propiedades de los árboles

Un árbol m-ario completo con  $i$  vértices internos contiene  $n = m \cdot i + 1$  vértices.

## Demostración

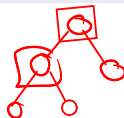
Puesto que el número de nodos internos es  $i$  y cada uno tiene  $m$  hijos diferentes de la raíz.

$$n = 2 \times 0 + 1$$

$$n = 1 \checkmark$$



$$n = 2 \times 1 + 1 = 3 \checkmark$$



$$n = 2 \times 2 + 1$$

$$n = 5$$

# Introducción a los árboles

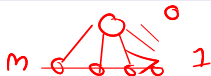
## Propiedades de los árboles

Un árbol  $m$ -ario de altura  $h$ , tiene máximo  $m^h$  hojas.

## Demostración

■ **Paso base** Con  $h = 1$  se tiene  $m$  hijos

■ **Paso inductivo** Tomando  $h$  cualquier se tienen  $m^h$  hijos. Para  $h + 1$  cada hijo tiene  $m$  hijos, por lo tanto, se tienen en total  $m * m^h = m^{h+1}$  hijos.



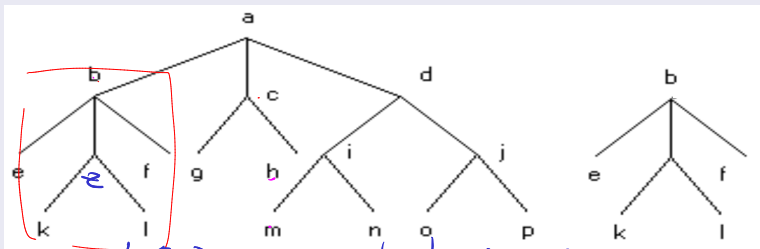
$$m \times m^h = m^{h+1}$$



# Introducción a los árboles

## Propiedades de los árboles

Un **subárbol** es un árbol que se obtiene al tomar uno de los nodos internos de un árbol como raíz.



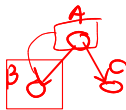
~~a, b, c, e, k, l, f, g, h, i, m, n, o, p~~  
~~e, b, k, z, l, f, a, g, c, h, m, i, n, d, o, j, p~~  
~~e, k, l, z, f, b, g, h, c, m, n, i, o, p, j, d, a~~

## Resumen

1) ¿Que es un árbol? Un grafo el cual no tiene CICLOS ni ARISTAS MULTIPLES

2) ¿Que es una raíz? Es un vértices que ha sido designado como origen del árbol y a todos los caminos hacia los otros vértices parte de él.

3) ¿Que es un padre? ¿Que es un hijo?



4) Antecesoros y descendientes



5) Arbol equilibrado: Los hojas están en una altura  $h$  o  $h-1$

6) Arbol m-ario completo: Todos los vértices internos TIENEN exactamente  $m$  hijos.

7) Arbol m-ario completo equilibrado: Cumplen 5 y 6

# Contenido

## 1 Introducción a los árboles

ADA) Implementación de árboles

## 2 Recorridos en árboles

## 3 Árboles de expansión

# Recorridos en árboles

## Definición

Los árboles con raíz se utilizan frecuentemente para almacenar información. Existen algoritmos de recorrido para visitar cada uno de los vértices para acceder a los datos. Los algoritmos más conocidos de recorrido de árboles son:

- Recorrido en preorden
- Recorrido en inorden
- Recorrido en postorden

# Recorridos en árboles

## Recorrido en preorden

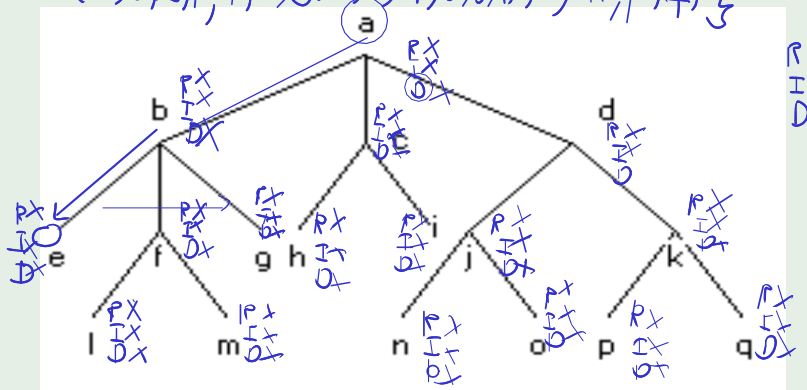
Para realizar el recorrido en preorden:

- 1 Visite la raíz. muestre la raíz
- 2 Visite los subárboles de izquierda a derecha (repita el procedimiento hasta llegar a las hojas)

# Recorridos en árboles

## Recorrido en preorden

$\{a, b, e, f, l, m, g, c, h, i, d, j, n, o, k, p, q, \}$



El recorrido preorden es  $a, b, e, f, l, m, g, c, h, i, d, j, n, o, k, p, q$

# Recorridos en árboles

## Recorrido en inorden

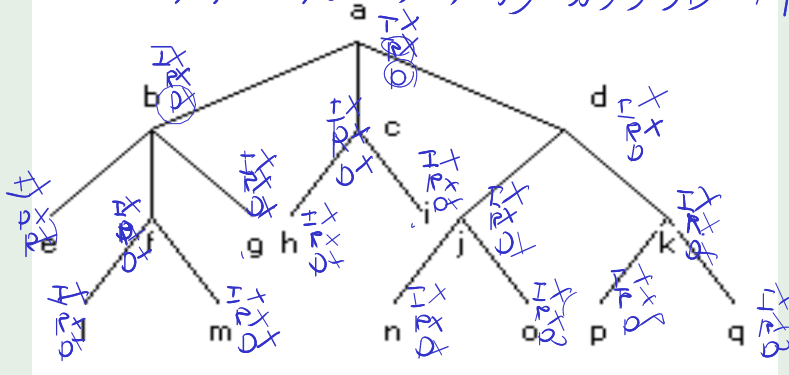
Para realizar el recorrido en inorden:

- 1 Visite el sub-arbol más izquierdo (primer hijo de izquierda a la derecha)
- 2 Visite la raíz. muestre la raíz
- 3 Visite los subárboles restantes de izquierda a derecha (repita el procedimiento hasta llegar a las hojas)

# Recorridos en árboles

## Recorrido en inorden

{ e, b, l, f, m, g, a, h, c, i, n, j, o, d, p, k, q }



El recorrido en inorden es: e, b, l, f, m, g, a, h, c, i, n, j, o, d, p, k, q



# Recorridos en árboles

## Recorrido en postorden

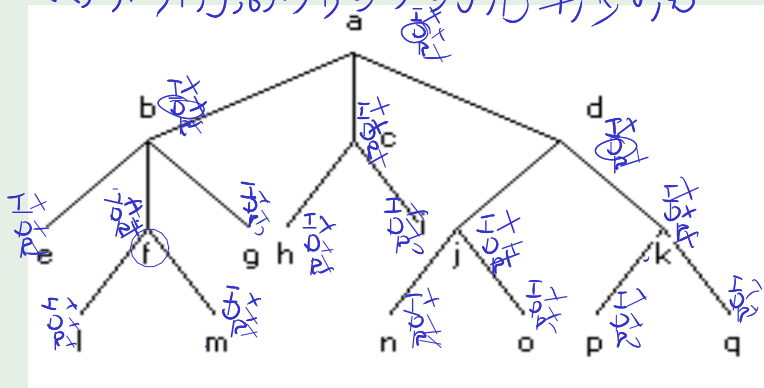
Para realizar el recorrido en postorden:

- 1 Visite el sub-arbol más izquierdo (primer hijo de izquierda a la derecha)
- 2 Visite los subárboles restantes de izquierda a derecha (repita el procedimiento hasta llegar a las hojas)
- 3 Visite la raíz. muestre la raíz

# Recorridos en árboles

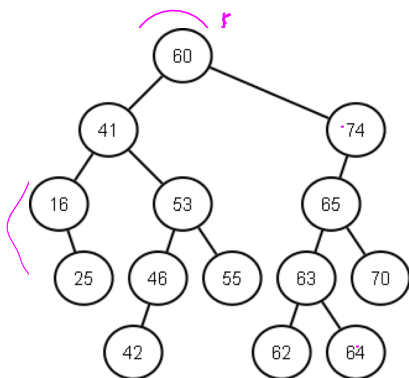
## Recorrido en postorden

{e, l, m, f, g, b, h, i, c, n, o, j, p, q, k, d, a}



El recorrido en postorden es:

e, l, m, f, g, b, h, i, c, n, o, j, p, q, k, d, a



Preorden

{60, 41, 16, 25, 53, 46, 42, 55, 74, 65, 63, 62, 64, 70}

Inorden

{25, 16, 41, 42, 46, 53, 55, 60, 62, 63, 64, 65, 70, 74}

Posorden

{25, 16, 42, 46, 55, 53, 41, 62, 64, 63, 70, 65, 74, 60}

Arbol biseccional de búsqueda

Recorrido preorden: 60 41 16 25 53 46 42 55 74 65 63 62 64 70

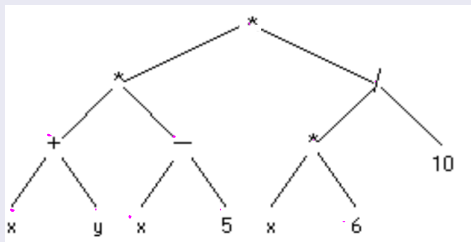
Recorrido inorden: 25 16 41 42 46 53 55 60 62 63 64 65 70 74

Recorrido posorden: 25 16 42 46 55 53 41 62 64 63 70 65 74 60

# Recorridos en árboles

## Expresiones aritméticas

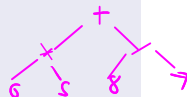
Las expresiones matemáticas pueden ser representadas usando árboles:



$5 + 7$



$(6 * 5) + (8 / 7)$



- ➔ ■ En preorden:  $(* (* (+ x y) (- x 5)) (/ (* x 6) 10))$
- ➔ ■ En inorden:  $((x + y) * (x - 5)) * ((x * 6) / 10)$
- ➔ ■ En postorden:  $((((x y +) (x 5 -) *) ((x 6 *) 10 /) *)$

# Contenido

1 Introducción a los árboles

2 Recorridos en árboles

3 Árboles de expansión

# Árboles de expansión

## Definición

Es un problema que está asociado a como obtener un árbol expansión para un grafo. Este árbol contiene todos los nodos del grafo y algunas de sus aristas para asegurar conectividad

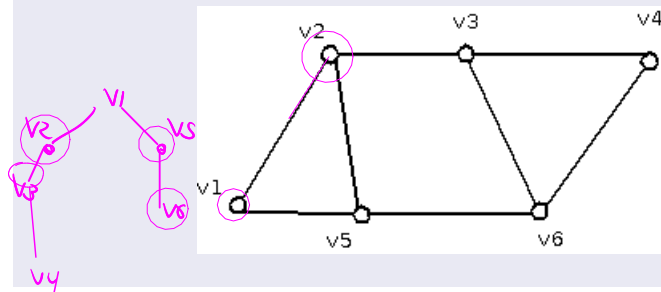
## Definition

Un árbol de expansión de un **grafo conexo**  $G=(V,E)$  es un árbol que tiene el conjunto de nodos  $N$  y es subgrafo de  $G$ . Esto es, un árbol de expansión es conexo, a cíclico y tiene a todo  $N$  y a parte de  $A$  como un conjunto de aristas.

# Árboles de expansión

## Obtención del árbol

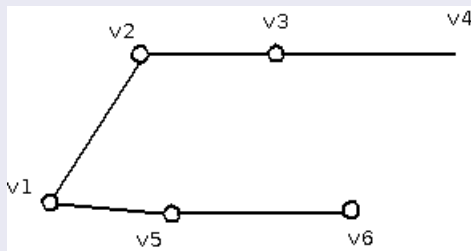
Hay muchas formas de obtener un árbol de expansión. Al empezar a borrar aristas para borrar los ciclos, por ejemplo:



# Árboles de expansión

## Obtención del árbol

Al eliminar  $\{(v1, v5), (v3, v6)(v4, v6)\}$



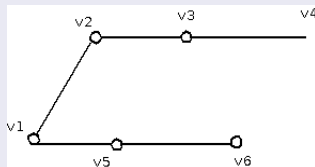


# Árboles de expansión

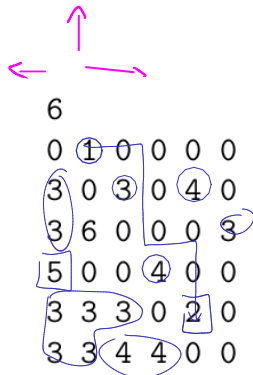
## Obtención del árbol

Una forma más sencilla es aplicar BFS (Búsqueda por amplitud) a un grafo evitando recorrer los nodos ya visitados, desde un nodo arbitrario, esto nos genera un árbol de expansión, para ejemplo anterior si comenzamos en  $V1$  obtenemos.

- $\{(v1, v2), (v1, v5)\}$
- $\{(v2, v3), (v5, v6)\}$
- $\{(v3, v4)\}$

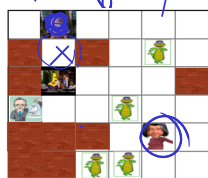


# Arbol de juego

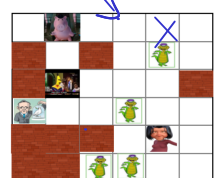
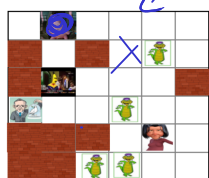


- pos Jogador  
 Reconidos  
 Cam no 11evon

0 1 2 3 4 5



Pierde



- 19
- 10
- 11
- 01
- 02
- 03
- 13
- 23
- 24
- 34
- 44



Kenneth H. Rosen.

*Discrete Mathematics and Its Applications.*

McGraw-Hill Higher Education, 7th edition, 2011.

Chapter 11. Graphs.

# Gracias

Próximo tema:  
Lenguajes y gramáticas