Matemáticas discretas II

Facultad de Ingeniería. Universidad del Valle

Marzo 2018

Contenido

Contenido

Universidad del Valle

Introducción a los árboles

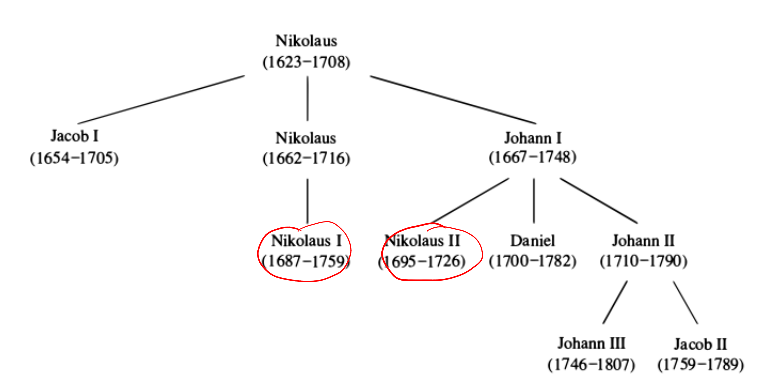
Introducción

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

Introducción a los árboles

Definición 1

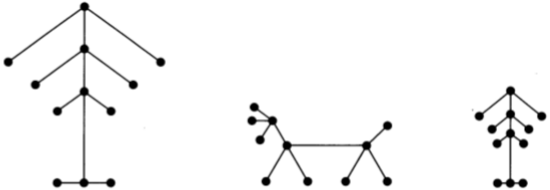
Un árbol no tiene ningún circuito simple. Esto quiere decir que no puede contener aristas múltiples ni ciclos.



Introducción a los árboles

Teorema 1

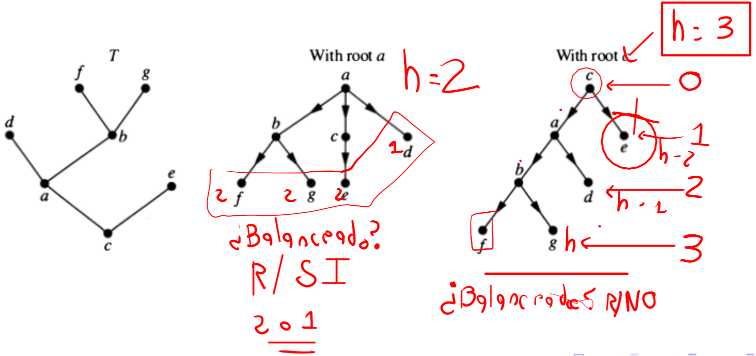
Un grado no dirigido es un árbol si y sólo si hay un único camino entre cada par de .



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 **raiz** hasta el vértices, pero excluyéndolo a él
- Los **descendientes** de un vértice son todos aquellos que tienen a v como antecesor

Definición 4

- Un vértice es llamado **hoja** si no tiene hijos
- Los vértices de los hijos son llamados **vértices internos**

Definición 5

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

Definición 6

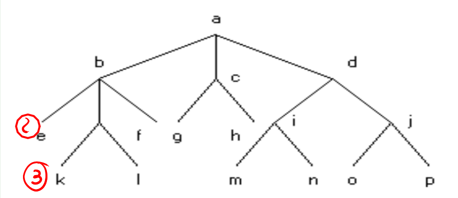
La **altura** de un árbol es la **longitud del camino más largo** desde la raíz hasta cualquier vértice

Definición 7

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

Introducción a los árboles

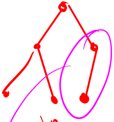
Ejemplo de árbol equilibrado



1. El árbol esta **equilibrado**
2. Su **altura** es 3
3. j es el **padre** de p
4. Los **antecesores** de n son $\{i, d, a\}$
5. los **descendientes** de d 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**

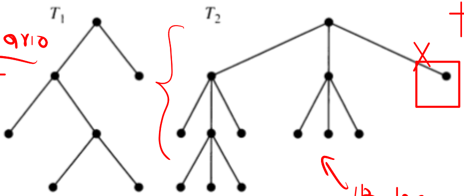
Definición 8

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.



¿Binario?
SI
¿Binario completo?
NO

binario



ternario completo

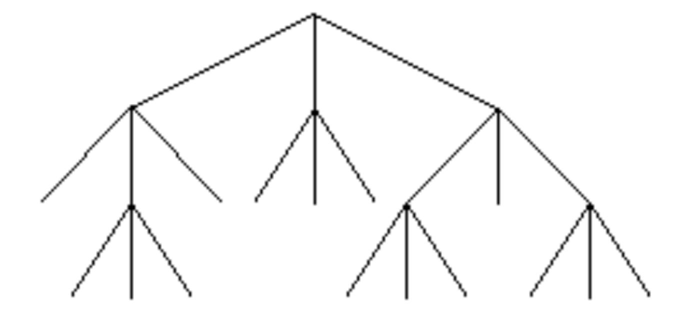
¿Balanceado? NO

1- Un arbol es m-ario si y sólo si cada nodo(vertice interno) tiene a lo máximo m hijos.
2- Un arbol es m-ario completo sí y sólo cada nodo (vértice interno) tiene EXACTAMENTE m hijos
3- Un arbol es balanceado si el nivel (altura) de sus hojas es h o h-1
4- La altura h de un árbol es el nivel (altura) de la hoja más baja.

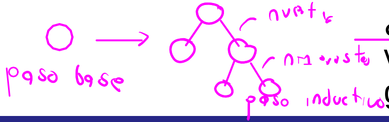
Definición 9

equilibrado

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



Introducción a los árboles



¿QUe implica agregar un vértice?. Es NECESARIO agre-gar una rista.

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

Universidad del Valle

Introducción a los árboles

Propiedades de los árboles

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

Demostración

Puesto que el número de nodos internos es $\boxed{i}$ y cada uno tiene *m* hijos diferentes de la raiz.x

Paso base

$m = 0$

$n = 0 \times i + 1$

$\boxed{n = 1}$

Paso inductivo

i nodos internos

$n = m \times i + 1 \longrightarrow m(i+1)+1$

+m nodos

$mi + m + 1$

Paso base, solo es la raiz, entonces el m = 0, la formula me dice que HAY UN SOLO VERTICES.

Paso inductivo, tengo un arbol con un m cualquiera m>0, e i padres, la formula me dice n = m*i+1 (Paso N)

(PASO N+1) Para agregar un vértice ¿Cuantos tengo que agregar? m ¿Porque? para que se conserve que sea completo. Entonces en total n+m vertices

¿Que pasa con el número i (vertices internos)? entonces cuando agrego un vértices este queda como i +1 aplicando la formula n' = m(i+1) + 1 --> n' = m*i+1 + m

$\boxed{n' = n + m}$ Vertices

Introducción a los árboles

Propiedades de los árboles

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

Demostración

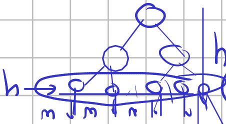
- **Paso base** Con $h = 1$ se tiene m hijos
- **Paso inductivo** Tomando h cualquier se tienen m^h hijos. Para $h + 1$ cada hijo tiene m hijos, por lo tanto, se tienen en total $m * m^h = m^{h+1}$ hijos.

Universidad del Valle

h tiene max m^h

Paso base $h=0$ $\quad m^0 = 1$ ✓

Paso inductivo $(h \to h+1)$



h: max m^h

$$\underbrace{(m+m+m+m+m)}_{m^h} \Big\} \; m\boxed{m^h}$$

Formula

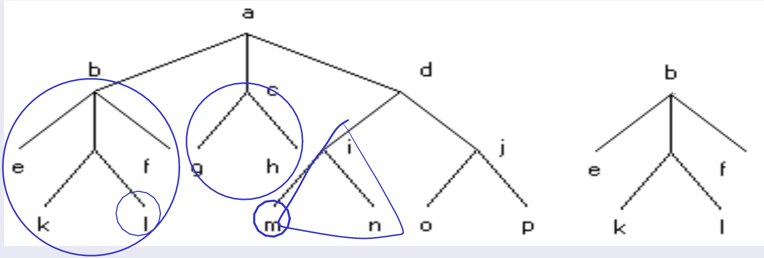
1) $m^{h+1} = m * m^h$

2) $m * m^h = m^{h+1}$

Lo que toca asumir es que tienes m^h hojas cuando al altura del árbol es h, si cada uno de estas hojas tiene m hijos en total tendremos m*m^h

Propiedades de los árboles

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



Universidad del Valle

Contenido

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:

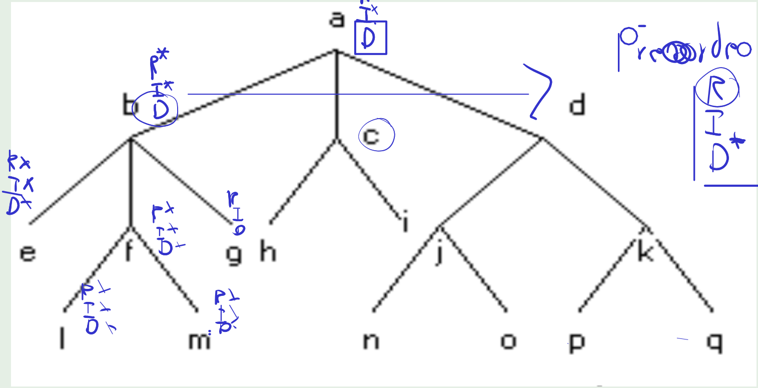
1. Visite la raíz. muestre la raíz
2. Visite los subárboles de izquierda a derecha (repita el procedimiento hasta llegar a las hojas)

Universidad del Valle

Recorridos en árboles

abeflmgchidjnokpq

Recorrido en preorden



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

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

Recorridos en árboles

I
R
D

Recorrido en inorden

Para realizar el recorrido en inorden:

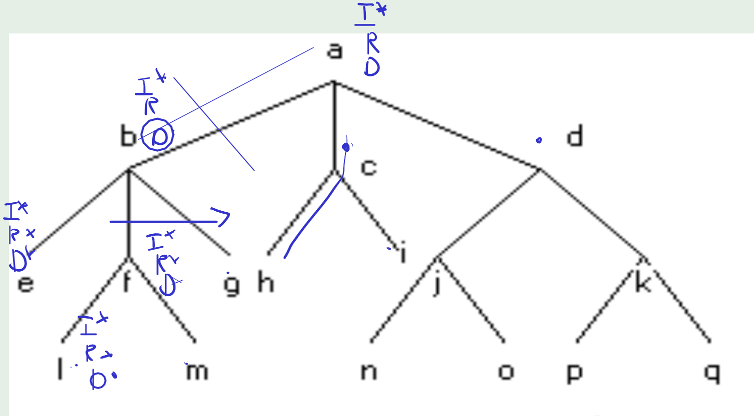
1. Visite el sub-arbol más izquierdo (primer hijo de izquierda a la derecha)
2. Visite la raíz. muestre la raíz
3. Visite los subárboles restantes de izquierda a derecha (repita el procedimiento hasta llegar a las hojas)

Universidad del Valle

Recorridos en árboles

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

Recorrido en inorden



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

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

Universidad del Valle

Recorridos en árboles

Recorrido en postorden

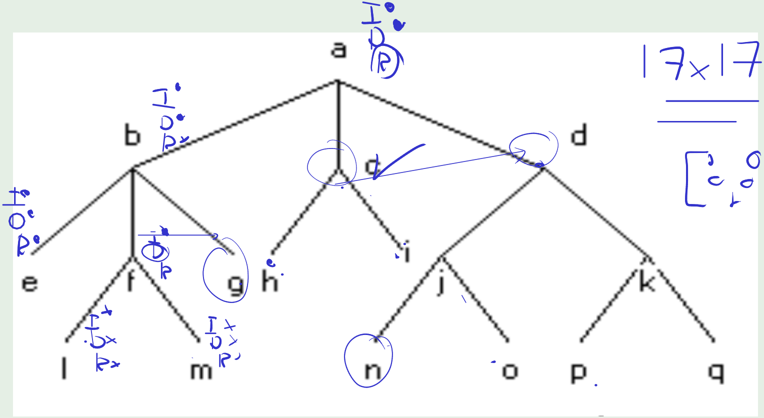
Para realizar el recorrido en postorden:

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

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

Recorrido en postorden



17×17

El recorrido en postorden es:

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

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

Preorden: Primero raiz, luego izquierda y finalmente derecha(s)
Inorden: Primero izquierda luego raiz y finalmente derecha
Posorden: Primero izquierda, luego derecha y finalmente raiz.

Representación de arboles en el computador

- Perfectamente puedo usar una matriz de adyacencia. Porque los arboles son unc lase de grafos. Pero, ¿Que problema ven?

```
(define-struct arbol (valor hizq hder))
(define-struct arbol3 (valor hizq hder1 hder1))
```

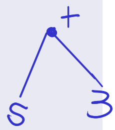
¿Que tipo de estructura eran? <-- Recursivas

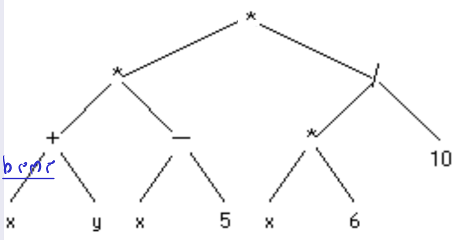
Recorridos en árboles

Expresiones aritméticas

Las expresiones matemáticas pueden ser representadas usando árboles:

5 + 3



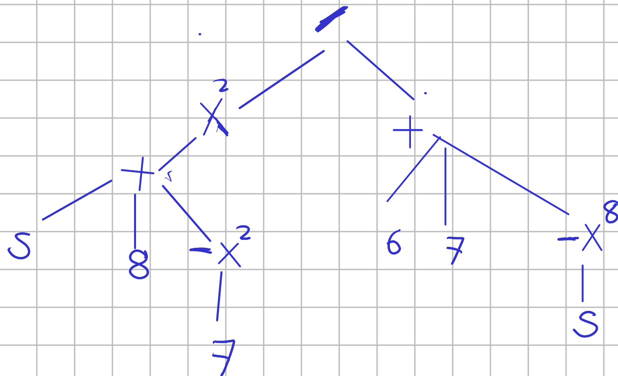


Racket = Scheme
Lisp

C++
Java
Ruby

- 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 /) *)

$$\frac{(5+8-7^2)^2}{(6+7-5^8)}$$



Preorden $\begin{smallmatrix}P\\I\\D\end{smallmatrix}$ (/ (sqr (+ 5 8 (- (sqr 7))))
(+ 6 7 (- (expt 5 8)))
)

Inorden $(5+8-7^2)^2 / (6+7-5^8)$

Posorden $(5 \ 8 \ 7 \ x^2 \ +) \ x^2$

$(6 \ 7 \ 5 \ x^8 \ +) \ /$

Contenido

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

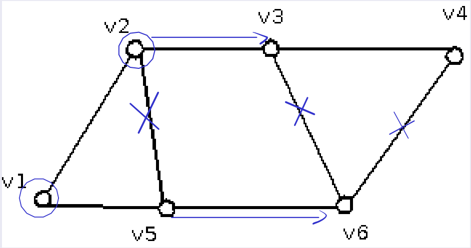
Un árbol de expansión de un **grafo conexo** G=(V,E) es un árbol que tiene el conjunto de nodos N y es subgrafo de G. Esto es, un árbol de expansión es conexo, a cíclico y tiene a todo N y a parte de A como un conjunto de aristas.

Universidad del Valle

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:



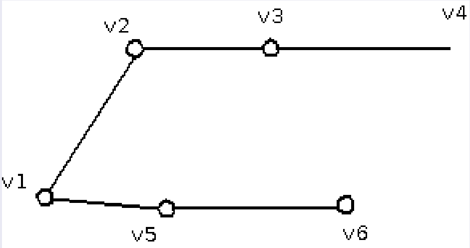
(v1 - v2) (v5 - v6)
(v1 - v5)
(v2 - v3)

Universidad del Valle

Árboles de expansión

Obtención del árbol

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

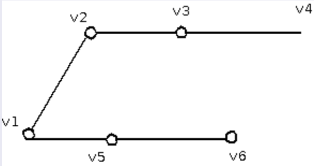


Árboles de expansión

Obtención del árbol

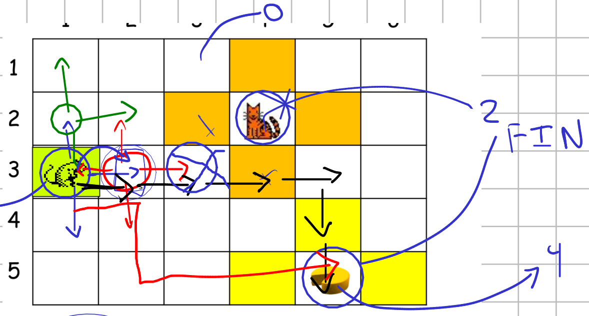
Una forma más sencilla es aplicar BFS (Búsqueda por amplitud) a un grafo evitando recorrer los nodos ya visitados, desde un nodo arbitrario, esto nos genera un árbol de expansión, para ejemplo anterior si comenzamos en V1 obtenemos.

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



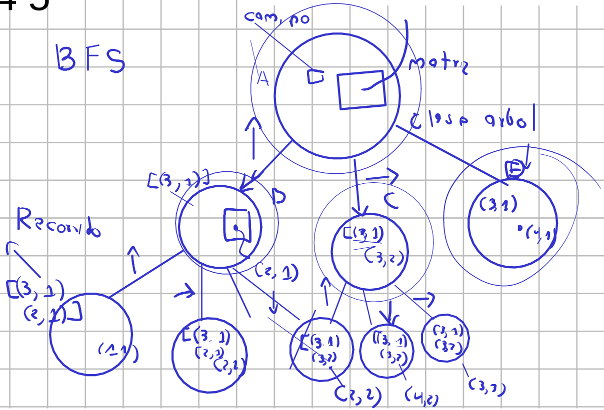
3 1
3 2
3 3
3 4
3 5
4 5

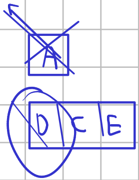
↑ ↓ → ←
Operadores



3 1
3 2
4 2
4 3
4 4
4 5
5 5

BFS





Referencias

Kenneth H. Rosen.
Discrete Mathematics and Its Applications.
McGraw-Hill Higher Education, 7th edition, 2011.
Chapter 11. Graphs.

Gracias

Próximo tema:
Lenguajes y gramáticas