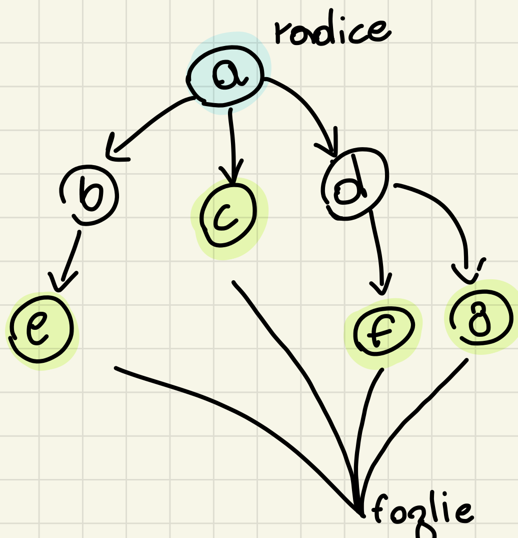
6

Un alberto é un tipo particolare di DAG

- ha un solo nodo sorgente detto radice
- un nodo che non sia radice ha un solo arco entrante
- i nodi pozzo sono chiamati foglie (nodi esterni)
- i nodi hanno un grado "parentale"



proprietá

- $K_{in} = 0$ solo per la radice, gli altri 1
- K_{out} no restrizioni
- per ogni nodo V ho un solo cammino tra la radice e V
- un albero non è mai vuoto
- se é finito esiste almeno una foglia
- i nodi intermedi sono sia padre che figlio
- se rappresento gerarchicamente le frecce sono implicite

Foreste e path

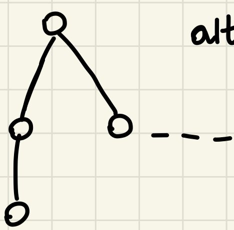
una foresta é un'unione disgiunta di alberi. (é un grafo)
Ogni albero ha la propria radice.

Un path è è un grafo un cui tutti i nodi hanno Kout=1 (tranne l'unica foglia)

Profonditá

è la lunghezza del cammino tra la radice e V

L'altezza é la profonditá massima (é una proprietá dell'intero albero)

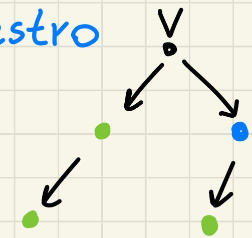


altezza = 2

profonditá = 1

Alberi binari

ogni nodo ha un massimo di due figli, i figli di un nodo V sono ordinati e detti figlio sinistro e figlio destro

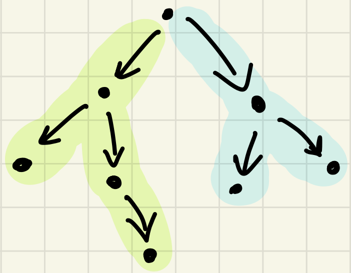


Proprietá

- ha al massimo 2^p nodi con profonditá p
- ha al massimo $\sum_{i=0}^{n} 2^i = 2^{n+1}-1$ nodi con altezza n

possiamo vederla come una struttura ricorsiva

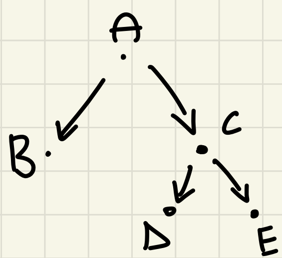
- un nodo radice
- un albero binario sinistro
- Un albero binario destro

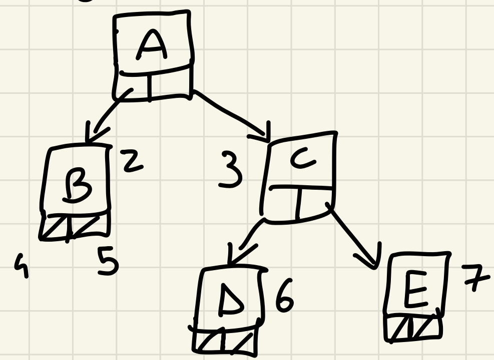


rappresentazione

1) collezione di nodi dove:
   - la radice é segnalata
   - ogni nodo ha due puntatori

2) tabella con $2^{n+1}-1$ righe, n = altezza albero

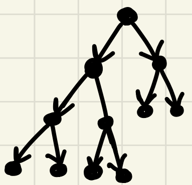




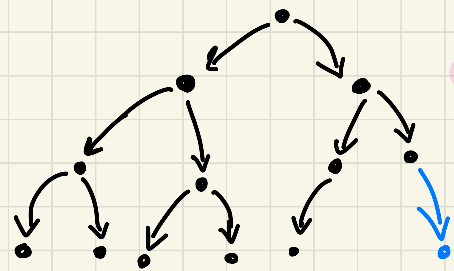
1	A	1
2	B	1
3	C	1
4	$\emptyset$	$\emptyset$
5	$\emptyset$	$\emptyset$
6	5	1
7	E	1

Tipi di alberi binari

- pieno o strettamente binario: ogni nodo interno ha due figli
- completo se ha altezza n, a ogni profondità i $0 \leq i < n$ ci sono 2^i



pieno ma non completo (ci sono degli ∅ nella tabella)

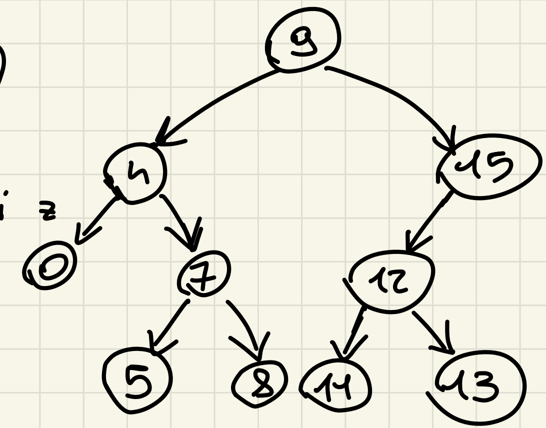


non pieno completo

non sarebbe completo

- bilanciato: se per ogni nodo V la differenza tra il numero di nodi dell'albero sinistro di V e quello destro é al massimo 1

Binary search tree (BST)



é un albero $G(V,E)$ con $V \subseteq \mathbb{Z}$ (insieme ordinabile)
t.c. per ogni nodo di di V

- ogni nodo dell'albero sinistro di z é minore di z
- destra maggiori

Attraversamento di un albero binario

- visitare <u>tutti</u> i nodi, di solito con ordine particolare
- un attraversamento che elenca un nodo solo una volta, é una enumerazione

tipi:

- in profonditá: ogni ramo fino in fondo
- in larghezza: prima i nodi vicino alla radice

Tipi di attraversamento in profonditá

sono basati sul quando enumero un nodo

notazione:

- V visito il nodo (enumero)
- L attraverso ricorsivamente il sottoalbero sinistro del nodo corrente
- R "" destro ""

1) Preordine VLR

2) Ordine LVR

3) Postordine LRV

(si puó implementare con una pila LIFO)

Enumerazione in ampiezza

Visito tutti i nodi ad una profonditá prima di scendere Coda FIFO

N. di foglie in un albero

Un albero finito ha sempre almeno una foglia.

Se ho n nodi interni in un albero pieno ho $n+1$ foglie

Chiusura riflessiva

relazione R su S

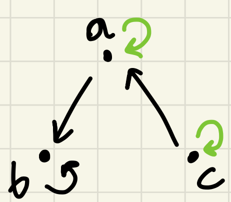
La chiusura riflessiva è la più piccola rel. riflessiva R^{rifl} su S che contiene R

$R^{rifl} = R \cup I_S \qquad R \subseteq R^{rifl}$

$S = \{a, b, c\} \quad R \subseteq S^2$

$R = \{<a, b>, <b, b>, <c, a>\}$

$$\begin{array}{c|ccc} & a & b & c \\ \hline a & \overset{1}{0} & 1 & 0 \\ b & 0 & 1 & 0 \\ c & 1 & 0 & \overset{1}{0} \end{array}$$



$R^{Rifl} = \{<a,a>, <a,b>. <c,a>, <b,b>, <c,c>\}$

Chiusura transitiva

La chiusura transitiva è la più piccola rel riflessiva R^{trans} su S che contiene R