

Arbori Generalizați

multime $\checkmark \rightarrow$ relație de precedență
 $i < j \quad a_i \prec a_j$

Propu. a) anti-reflexivitate ~~$a_i \prec a_i$~~

b) trans. $\left. \begin{array}{l} a \prec b \\ b \prec c \end{array} \right\} \rightarrow a \prec c$

c) $\left. \begin{array}{l} \nexists a \in \checkmark \\ \nexists b \in \checkmark \\ a \neq b \end{array} \right\} \rightarrow \begin{array}{l} a \prec b \\ \text{sau} \\ b \prec a \end{array}$

• relația de precedență nu det.
bucle închise în \checkmark

Pt. vectori \rightarrow relație de precedență completă.

- vector \rightarrow fiecare el. are un succesor
- arbore \rightarrow numără fiecare de succesorii
2 el. distincte nu au
același succesor

Subarbore: \rightarrow oricare nod este
rădăcina a unui
arbore parțial

Arbore Parțial \rightarrow nu întotdeauna un
subarbore

Nivel \rightarrow nivel 0 \rightarrow rădăcină
 \rightarrow nivel 1 \rightarrow rădăcină

nivel max \rightarrow înălțimea arborelui

Gradul \rightarrow nr. de fii

grad 0 \rightarrow frunză
nod terminal

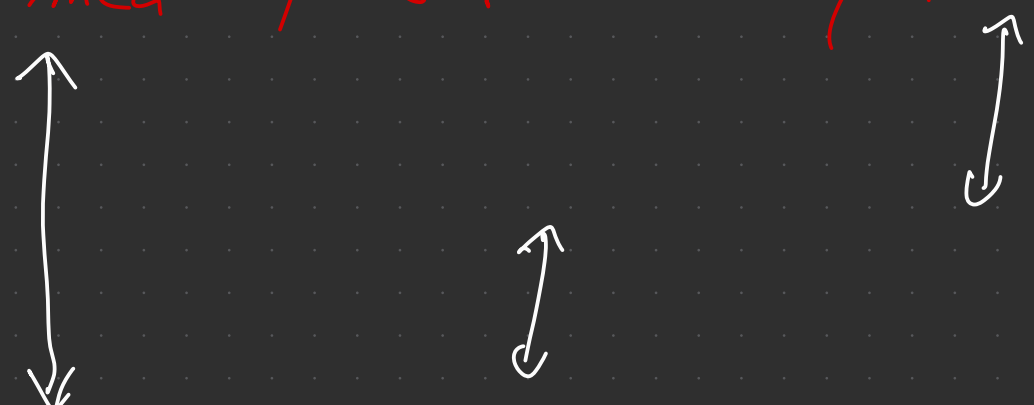
Drum / Cale \rightarrow secvență
 \rightarrow lungimea



Înălțimea are lungimea drumului 1

- tatăl unui nod este strămoșul sau direct (predecesor)
- fiul unui nod \rightarrow descendent direct (succesor)
- înălțimea nu are strămoș propriu
- frunzele nu au descendent propriu

Înălțimea / În. unui nod / Adâncimea

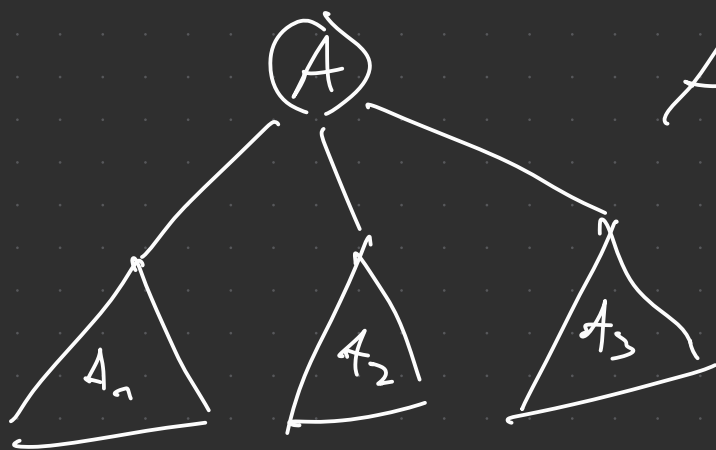


Percurși + căutări < adâncime (depth)
cuprindere (breadth)

adâncime < preordine RSD
in SRD
post SDR

A nul \rightarrow percurși \rightarrow lista vidă

A un nod \rightarrow toate 3 perc. \rightarrow la fel



A răd.

A_1, A_2, \dots, A_k
subarbori

Preordine : R $\rightarrow A_1, A_2, \dots, A_k$

Inordine : $A_1 \rightarrow$ R $\rightarrow A_2 \dots A_k$

Post : $A_1 \dots A_k \rightarrow$ R

Implementare

Tablouri \rightarrow vector de tat_i / indicator de părinte

Index	0	1								
Val	0	A	B	C	D	E	F	G	H	I
părinte	0	0								

reserved \rightarrow errors

Prim fin / frate chepta

Index	
Val	
Prim fin	
Frata chepta	

Implementare cu Liste

→ fiecare nod are o lista → pointer către
frății lui

→ dynamic

→ static (array Implementation)