

Bináris fa láncolt ábrázolása:

Két pointeres csúcs. Node

Node
+ $key : \mathcal{T} // \mathcal{T}$ valamilyen ismert típus
+ $left, right : Node^*$
+ $Node() \{ left := right := \emptyset \}$ // egycsúcsú fát képez belőle
+ $Node(x : \mathcal{T}) \{ left := right := \emptyset ; key := x \}$

Három pointeres csúcs: Node3

Node3
+ $key : \mathcal{T} // \mathcal{T}$ valamilyen ismert típus
+ $left, right, parent : Node3^*$
+ $Node3(p : Node3^*) \{ left := right := \emptyset ; parent := p \}$
+ $Node3(x : \mathcal{T}, p : Node3^*) \{ left := right := \emptyset ; parent := p ; key := x \}$

Ismétlésképpen írjuk fel a három rekurzív algoritmust, ahogyan a jegyzetben szerepel:

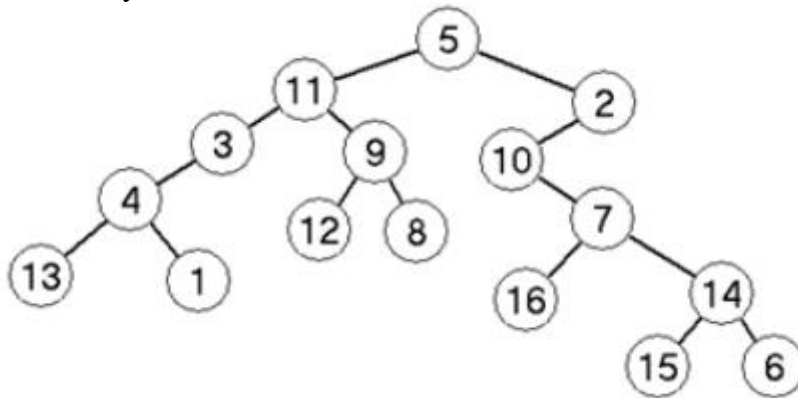
BinTree – absztrakt bináris fa típus, $t \rightarrow left$ és $t \rightarrow right$ helyett szokás $left(t)$ és $right(t)$ jelölést is használni. Az üres fát szokták Ω -val jelölni, így $t = 0$ helyett $t = \Omega$ is használható. Láncolt ábrázolású bináris fák esetén a bejáró algoritmusok paramétere lehetne $t : Node^*$, vagy $t : Node3^*$.

$(preorder(t : BinTree))$	$(inorder(t : BinTree))$	$(postorder(t : BinTree))$
$t \neq \emptyset$	$t \neq \emptyset$	$t \neq \emptyset$
process(t)	inorder($t \rightarrow left$)	postorder($t \rightarrow left$)
preorder($t \rightarrow left$)	process(t)	postorder($t \rightarrow right$)
preorder($t \rightarrow right$)	inorder($t \rightarrow right$)	process(t)
SKIP	SKIP	SKIP

Megjegyzés: A jegyzetben megtalálható a preorder és az inorder bejárás egy másik alakja: az utolsó rekurzív hívást ciklus helyettesíti – így hatékonyabb.

1. feladat

Egy konkrét bináris fa bejárása a három tanult rekurzív algoritmussal, ha a feldolgozás a kulcs kiírása, milyen sorrendben írná ki a kulcsokat?



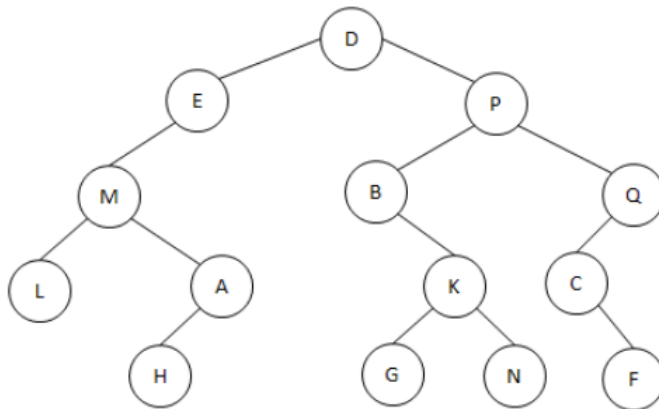
2. feladat

Egy nem teljes bináris fa preorder + inorder, vagy postorder + inorder bejárásából rekonstruáljuk, hogyan nézett ki a fa. Miért nem lehet rekonstruálni a preorder+postorder bejárásból?

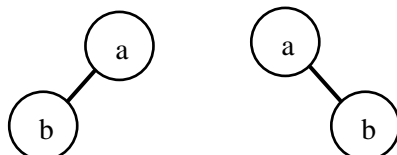
PREORDER: D E M L A H P B K G N Q C F

INORDER: L M H A E D B G K N P C F Q

Megoldás:



Miért nem jó a preorder és postorder? Ha egy-gyerekes a csúcs, nem lehet tudni belőle, hogy az egy-gyerek melyik irányban van:



PRE: a b

POST: b a

a b

b a