



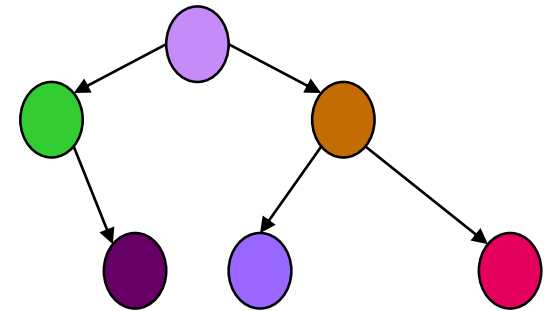
Fák, bináris fák



# Bináris fa



A fa (bináris fa) rekurzív adatszerkezet:

$$\text{BinFa} := \begin{cases} \text{ÜresFa} \\ \text{Rekord}(\text{Elem}, \text{BinFa}, \text{BinFa}) \end{cases}$$
$$\text{Fa} := \begin{cases} \text{ÜresFa} \\ \text{Rekord}(\text{Elem}, \text{Fá}^k) \end{cases}$$




# Bináris fa



A fa (bináris fa) rekurzív adatszerkezet ábrázolása:

**Típus** TBinfa=**Rekord**

(elem: TElem

bal, jobb: TBinfa)

**Típus** TFa=**Rekord**

(elem: TElem

ágak: **Sorozat** (TFa) )

A fa elemek sokasága, ezért szükség lesz:

- kezdő- (gyökér) elemre
- aktuális elemre





# Bináris fa – dinamikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Üres: Binfa

üres? (Binfa) : Logikai

egyeleműfa (Elem) : Binfa

Balrailleszt (Binfa, Binfa) :  $\text{Binfa} \cup \{\text{NemDef}\}$

előfeltétel: Binfa nem üres, a bal része üres

Jobbrailleszt (Binfa, Binfa) :  $\text{Binfa} \cup \{\text{NemDef}\}$

előfeltétel: Binfa nem üres, a jobb része üres

gyökérelém (Binfa) :  $\text{Elem} \cup \{\text{NemDef}\}$

előfeltétel: Binfa nem üres





# Bináris fa – dinamikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

balgyerek (Binfa) :  $\text{Binfa} \cup \{\text{Nemdef}\}$

előfeltétel: Binfa nem üres

jobbgyerek (Binfa) :  $\text{Binfa} \cup \{\text{Nemdef}\}$

előfeltétel: Binfa nem üres

gyökérmódosít (Binfa, Elem) :  $\text{Binfa} \cup \{\text{Nemdef}\}$

előfeltétel: Binfa nem üres







# Bináris fa – dinamikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Üres (bf) :

bf := sehova

Eljárás vége.

üres? (bf) :

üres? := bf = sehova

Függvény vége.

egyeleműfa (Elem) :

Lefoglal (bf, (Elem, sehova, sehova))

egyeleműfa := bf

Függvény vége.





# Bináris fa – dinamikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Balrailleszt ( $bf, rf$ ) :

$bf \rightarrow bal := rf$

Eljárás vége.

Jobbrailleszt ( $bf, rf$ ) :

$bf \rightarrow jobb := rf$

Eljárás vége.

gyökérelém ( $bf$ ) :

$gyökérelém := bf \rightarrow elem$

Függvény vége.





# Bináris fa – dinamikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

balgyerek (bf) :

balgyerek := bf -> bal

Függvény vége.

jobbgyerek (bf) :

jobbgyerek := bf -> jobb

Függvény vége.

Gyökérmódosít (bf, Elem) :

bf -> elem := Elem

Eljárás vége.







# Bináris fa – statikus láncolás



Koncepció:

**Típus** TBinfa=**Rekord**

(gyökér, akt, szabad: 0..Max,  
t: tömb(1..Max, TBinfaelem))

Tbinfaelem=**Rekord**

(elem: TElem, bal, jobb: 0..Max)





# Bináris fa – statikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Üres: Binfa

üres? (Binfa) : Logikai

Egyeleműfa (Elem) : Binfa

Balrailleszt (Binfa, Elem) :  $\text{Binfa} \cup \{\text{NemDef}\}$

előfeltétel: Binfa aktuális eleme nem üres, a bal része üres

Jobbra illeszt (Binfa, Elem) :  $\text{Binfa} \cup \{\text{NemDef}\}$

előfeltétel: Binfa aktuális eleme nem üres, a jobb része üres

Gyökérre (Binfa) : Binfa





# Bináris fa – statikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

$\text{elem}(\text{Binfa}) : \text{Elem} \cup \{\text{NemDef}\}$

előfeltétel: Binfa aktuális eleme nem üres

$\text{Balra}(\text{Binfa}) : \text{Binfa} \cup \{\text{Nemdef}\}$

előfeltétel: Binfa aktuális eleme nem üres

$\text{Jobbra}(\text{Binfa}) : \text{Binfa} \cup \{\text{Nemdef}\}$

előfeltétel: Binfa aktuális eleme nem üres

$\text{Módosít}(\text{Binfa}, \text{Elem}) : \text{Binfa} \cup \{\text{Nemdef}\}$

előfeltétel: Binfa aktuális eleme nem üres





# Bináris fa – statikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Üres (bf) :

```
bf.gyökér:=0; bf.akt:=0; bf.szabad:=1
```

Eljárás vége.

üres? (bf) :

```
üres?:=bf.gyökér=0
```

Függvény vége.

Egyeleműfa (bf, Elem) :

```
bf.gyökér:=1; bf.akt:=1; bf.t(1):=(Elem, 0, 0)
```

Eljárás vége.

Gyökérre (bf) :

```
bf.akt:=bf.gyökér
```

Eljárás vége.





# Bináris fa – statikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Balrailleszt (bf, Elem) :

```
x:=bf.szabad; bf.akt:=x; bf.t(bf.akt).bal:=x  
bf.t(x):=(Elem, 0, 0); bf.szabad:=bf.szabad+1
```

Eljárás vége.

Jobbrailleszt (bf, Elem) :

```
x:=bf.szabad; bf.akt:=x; bf.t(bf.akt).jobb:= x  
bf.t(x):=(Elem, 0, 0); bf.szabad:=bf.szabad+1
```

Eljárás vége.

elem(bf) :

```
elem:=bf.t(bf.akt).elem
```

Függvény vége.







# Bináris fa – statikus láncolás



A Bináris fa rekurzív adatszerkezet műveletei:

Balra (bf) :

```
bf.akt:=bf.t (bf.akt) .bal
```

Függvény vége.

Jobbra (bf) :

```
bf.akt:=bf.t (bf.akt) .jobb
```

Függvény vége.

Módosít (bf, Elem) :

```
bf.t (bf.akt) .elem:=Elem
```

Eljárás vége.





# Bináris fa – statikus láncolás



Módosítás: a bináris fa egyes elemei tartalmazhatják a fölöttük levő elem azonosítóját is.

**Típus** TBinfa=**Rekord**

(gyökér, akt: 0..Max,

t: tömb(1..Max, TBinfaelem) )

Tbinfaelem=**Rekord**

(elem: TElem,

bal, jobb, szülő: 0..Max)





# Bináris fa – dinamikus láncolás



Módosítás: a bináris fa egyes elemei tartalmazhatják a fölöttük levő elem azonosítóját is – dinamikus láncolással.

**Típus** TBinfa=**Rekord**

(gyökér, akt: TBinfaelem'Mutató)

TBinfaelem=**Rekord**

(elem: TElem,

bal, jobb, szülő: TBinfaelem'Mutató)





# Bináris fa – új műveletek



Fa bejárások: Bal-közép-jobb (inorder)

BKJ (bf) :

Ha nem üres? (bf) akkor

BKJ (balgyerek (bf) )

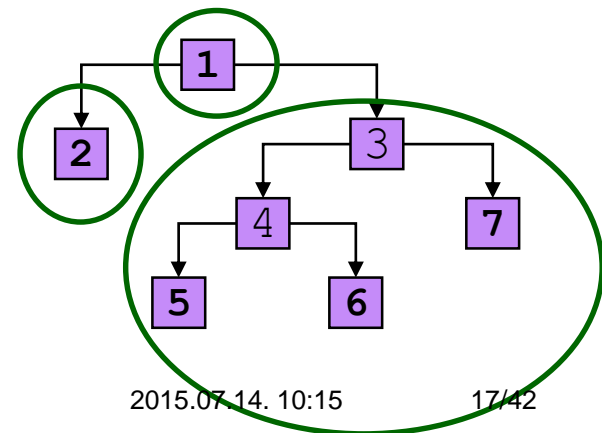
Ki: gyökérelém (bf)

BKJ (jobbgyerek (bf) )

Elágazás vége

Eljárás vége.

Alkalmazás: rendezőfa





# Bináris fa – új műveletek



Fa bejárások: Közép-bal-jobb (preorder)

KBJ (bf) :

Ha nem üres? (bf) akkor

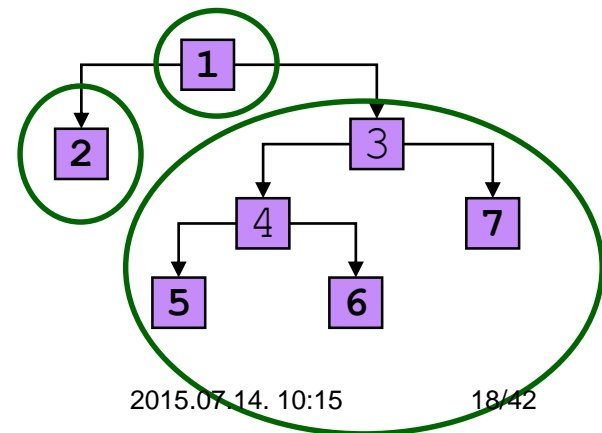
Ki: gyökérelem (bf)

KBJ (balgyerek (bf) )

KBJ (jobbgyerek (bf) )

Elágazás vége

Eljárás vége.







# Bináris fa – új műveletek



Fa bejárások: Bal-jobb-közép (postorder)

BJK (bf) :

Ha nem üres? (bf) akkor

BJK (balgyerek (bf) )

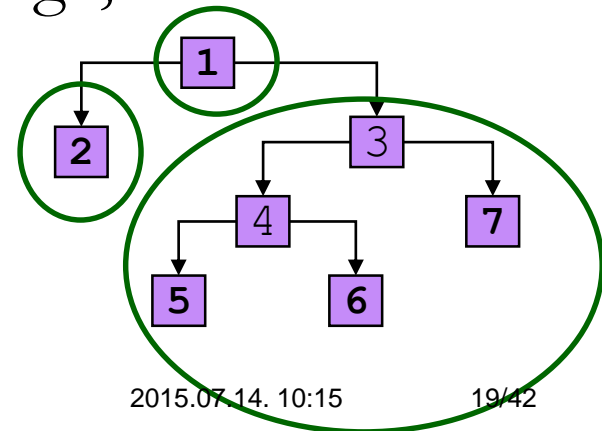
BJK (jobbgyerek (bf) )

Ki: gyökérelem (bf)

Elágazás vége

Eljárás vége.

Alkalmazás: Fa törlése, elemszáma, magassága, ...





# Bináris fa – új műveletek



## Fa törlése (dinamikus láncolással)

Törlés (bf) :

Ha nem üres? (bf) akkor

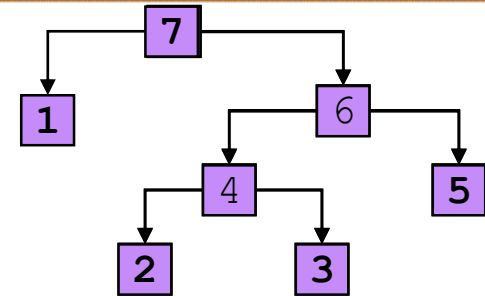
Törlés (balgyerek (bf) )

Törlés (jobbgyerek (bf) )

Gyökértörlés (bf)

Elágazás vége

Eljárás vége.



Gyökértörlés (bf) :

Felszabadít (bf) ; bf:=sehova

Eljárás vége.





# Kereső- és rendezőfák

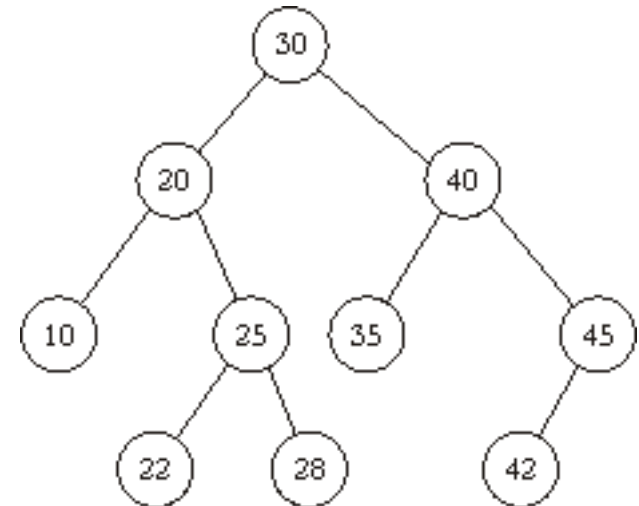


Közös tulajdonságok:

- A gyökérelem (vagy kulcsértéke) nagyobb vagy egyenlő minden tőle balra levő elemnél.
- A gyökérelem (vagy kulcsértéke) kisebb vagy egyenlő minden tőle jobbra levő elemnél.

Keresőfa specialitása:

- Nincs két azonos (kulcsú) elem.





# Keresőfák



Keresés ( $k, kf$ ) :

Elágazás

Üres? ( $kf$ ) esetén  $Keresés := kf$

$k < \text{gyökérelém}(kf) . \text{kulcs}$

esetén  $Keresés := Keresés(k, \text{balgyerek}(kf))$

$k > \text{gyökérelém}(kf) . \text{kulcs}$

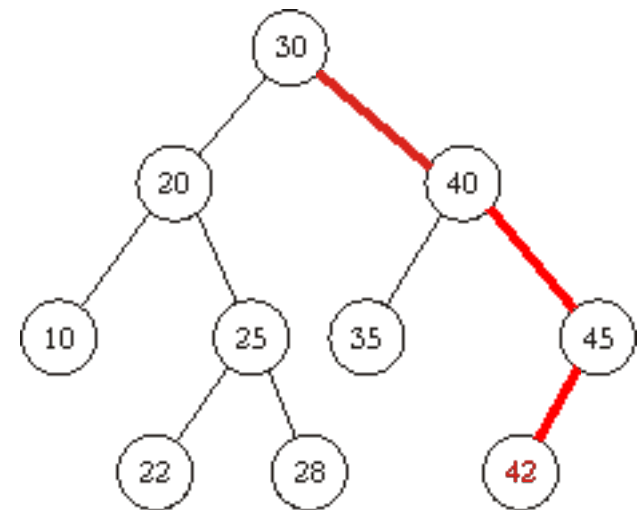
esetén  $Keresés := Keresés(k, \text{jobbgyerek}(kf))$

$k = \text{gyökérelém}(kf) . \text{kulcs}$

esetén  $Keresés := kf$

Elágazás vége

Függvény vége.





# Keresőfák



Beillesztés (e, kf) :

Elágazás

Üres? (kf) esetén  $kf := \text{egyeleműfa}(e)$

$e.\text{kulcs} < \text{gyökérelém}(kf) . \text{kulcs}$

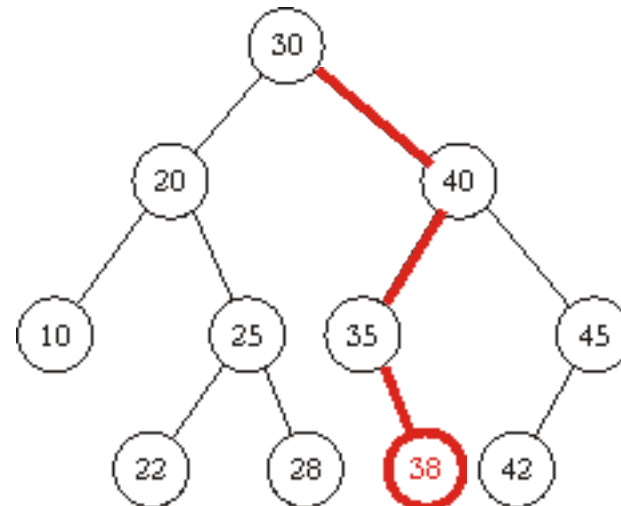
esetén  $\text{Beillesztés}(e, \text{balgyerek}(kf))$

$e.\text{kulcs} > \text{gyökérelém}(kf) . \text{kulcs}$

esetén  $\text{Beillesztés}(e, \text{jobbgyerek}(kf))$

Elágazás vége

Függvény vége.







# Rendezőfák



Beillesztés (e, kf) :

Elágazás

Üres? (kf) esetén  $kf := \text{egyeleműfa}(e)$

$e.\text{kulcs} \leq \text{gyökérelém}(kf) .\text{kulcs}$

esetén  $\text{Beillesztés}(e, \text{balgyerek}(kf))$

$e.\text{kulcs} > \text{gyökérelém}(kf) .\text{kulcs}$

esetén  $\text{Beillesztés}(e, \text{jobbgyerek}(kf))$

Elágazás vége

Függvény vége.





# Bináris fák



Néhány speciális fa művelet:

elemszám( $bf$ ) :

Ha üres?( $bf$ ) akkor elemszám:=0

különben elemszám:=1+elemszám(balgyerek( $bf$ )) +  
elemszám(jobbgyerek( $bf$ ))

Függvény vége.

magasság( $bf$ ) :

Ha üres?( $bf$ ) akkor magasság:=0

különben magas:=1+max(magasság(balgyerek( $bf$ )),  
magasság(jobbgyerek( $bf$ )))

Függvény vége.





# Bináris fák



Néhány speciális fa művelet:

levélszám (bf) :

Ha üres? (bf) akkor levélszám := 0

különben ha üres (balgyerek (bf)) és  
üres (jobbgyerek (bf))

akkor levélszám := 1

különben levélszám := levélszám (balgyerek (bf)) +  
levélszám (jobbgyerek (bf))

Függvény vége.





# Bináris fák



Leghosszabb út hossza (magasság számolásával):

$legh(bf, m)$  :

Ha  $\text{üres}(\text{balgyerek}(bf))$  és  $\text{üres}(\text{jobbgyerek}(bf))$   
akkor  $m := 1$ ;  $legh := 0$

különben  $x := legh(\text{balgyerek}(bf), mx)$

$y := legh(\text{jobbgyerek}(bf), my)$

*Ha  $mx > my$  akkor  $m := mx$  különben  $m := my$*

Ha  $x < y$  akkor  $z := y$  különben  $z := x$

Ha  $mx + my + 2 > z$  akkor  $legh := mx + my + 2$

különben  $legh := z$

Függvény vége.





# Bináris fák



szélesség (bf, szint) :

Ha nem üres? (bf)

akkor  $sz(szint) := sz(szint) + 1$

szélesség (balgyerek (bf), szint+1)

szélesség (jobbgyerek (bf), szint+1)

Eljárás vége.

széles (bf) :

szélesség (bf, 0); max:=0;

Ciklus i=1-től magas (fb) -ig

Ha  $sz(i) > sz(max)$  akkor max:=i

Ciklus vége

Eljárás vége.





# Nem bináris fák



A fa rekurzív adatszerkezet ábrázolása:

**Típus** TFa=**Rekord**

(elem: TElem

ágak: **Sorozat** (TFa) )

Egy speciális változat (ágszám felső korláttal):

**Típus** TFa=**Rekord**

(elem: TElem

db: Egész

ág: **Tömb** (1..Max, TFa) )







# Nem bináris fák



A fa rekurzív adatszerkezet műveletei:

Üres : Fa

üres? (Fa) : Logikai

Egyeleműfa (Elem) : Fa

gyerekszám (Fa) : Egész

Beilleszt (Fa, Fa) :  $Fa \cup \{NemDef\}$

Gyökérelem (Fa) :  $Elem \cup \{NemDef\}$

Gyökérmódosít (Fa, Elem) :  $Fa \cup \{NemDef\}$

Gyerek (Fa, Egész) :  $Fa \cup \{NemDef\}$





# Nem bináris fák



A fa műveletei megvalósítása dinamikus láncolással:

Üres (f) :

f := sehova

Eljárás vége.

üres? (f) :

üres? := (f = sehova)

Függvény vége.

Egyeleműfa (e) :

Lefoglal (f, (e, Üressorozat)); Egyeleműfa := f

Függvény vége.





# Nem bináris fák



A fa műveletei megvalósítása dinamikus láncolással:

gyerekszám (f) :

gyerekszám := Elemszám (f → ágak)

Függvény vége.

Gyökérelém (f) :

Gyökérelém := f → elem

Függvény vége.

Gyökérmódosít (f, e) :

f → elem := e

Eljárás vége.

*gyerekszám (f) :*

*gyerekszám := f → db*

*Függvény vége.*





# Nem bináris fák



Sorozattal ábrázolva:

Beilleszt (mire, mit) :

mire- $\rightarrow$ ágak := Végére (mire- $\rightarrow$ ágak, mit) \*

Eljárás vége.

Gyerek (f, i) :

Ha  $i \leq \text{ElemSzám}(f \rightarrow \text{ágak})$

akkor Gyerek :=  $f \rightarrow \text{ágak}(i)$

különben Gyerek := Üres

Függvény vége.



\*: Elejére is lehet, ha a sorrend nem fontos!



# Nem bináris fák



Tömbbel ábrázolva:

Beilleszt (mire, mit) :

mire  $\rightarrow$  db := mire  $\rightarrow$  db + 1

mire  $\rightarrow$  ág (mire  $\rightarrow$  db) := mit

Eljárás vége.

Gyerek (f, i) :

Ha  $i \leq f \rightarrow$  db

akkor Gyerek := f  $\rightarrow$  ág (i)

különben Gyerek := Üres

Függvény vége.





# Nem bináris fák – alkalmazás



## A fa elemszáma

elemszám(f) :

Ha üres?(f) akkor elemszám:=0

különben s:=1

Ciklus i=1-től f->db-ig

s:=s+elemszám(f->ág(i))

Ciklus vége

elemszám:=s

Függvény vége.







# Nem bináris fák – alkalmazás



## A fa magassága

magasság(f) :

Ha üres?(f) akkor magasság:=0

különben max:=0

Ciklus i=1-től f->db-ig

m:=magasság(f->ág(i))

Ha m>max akkor max:=m

Ciklus vége

magasság:=max+1

Függvény vége.

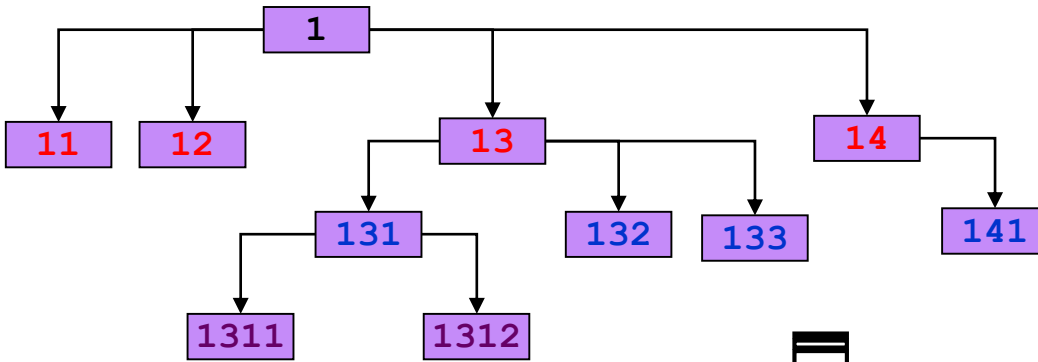




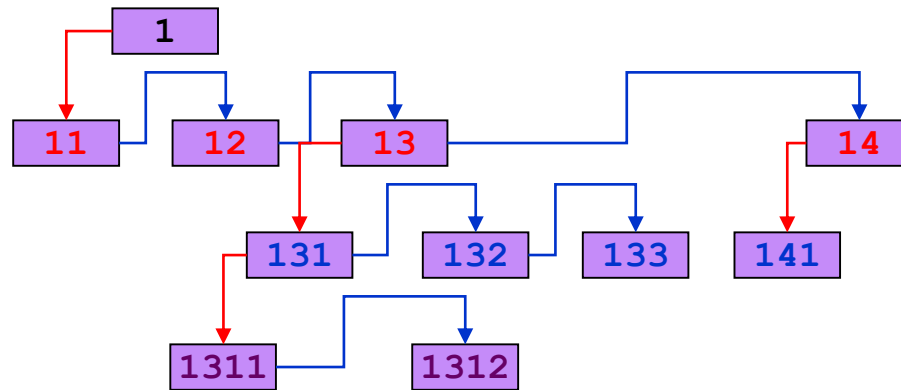
# Nem bináris fák – binárisan



Bináris ábrázolás:



balra az 1. gyerek  
jobbra a következő  
testvér





# Nem bináris fák – binárisan



Gyerek( $f, i$ ):

Ha  $i=1$  akkor Gyerek:= $f \rightarrow$ bal

különben Gyerek:=Testvér( $f \rightarrow$ bal,  $i$ )

Függvény vége.

Testvér( $f, i$ ):

Ha  $i=2$  akkor Testvér:= $f \rightarrow$ jobb

különben Testvér:=Testvér( $f \rightarrow$ jobb,  $i-1$ )

Függvény vége.





# Nem bináris fák – binárisan



Elsőgyerek (f) :

Elsőgyerek := f -> bal

Függvény vége.

Következőgyerek (gy) :

Következőgyerek := gy -> jobb

Függvény vége.

Vanméggyerek (gy) :

Vanméggyerek := (gy -> jobb ≠ se hova)

Függvény vége.





# Nem bináris fák – binárisan



Gyerek (f, i) :

gy := Elsőgyerek (f)

Ha  $i=1$  akkor Gyerek := gy

különben Ciklus  $j=2$ -től  $i$ -ig

gy := Következőgyerek (gy)

Ciklus vége

Gyerek := gy

Függvény vége.





# Nem bináris fák – alkalmazás



A fa magassága

magasság(f) :

Ha üres?(f) akkor magasság:=0

különben gy:=elsőgyerek(f)

max:=magasság(gy)

Ciklus amíg van méggyerek(gy)

gy:=következőgyerek(gy)

m:=magasság(gy)

Ha m>max akkor max:=m

Ciklus vége

magasság:=max+1

Függvény vége.







Fák, bináris fák