OBJEKTUM ORIENTÁLT PROGRAMOZÁS

Python nyelven

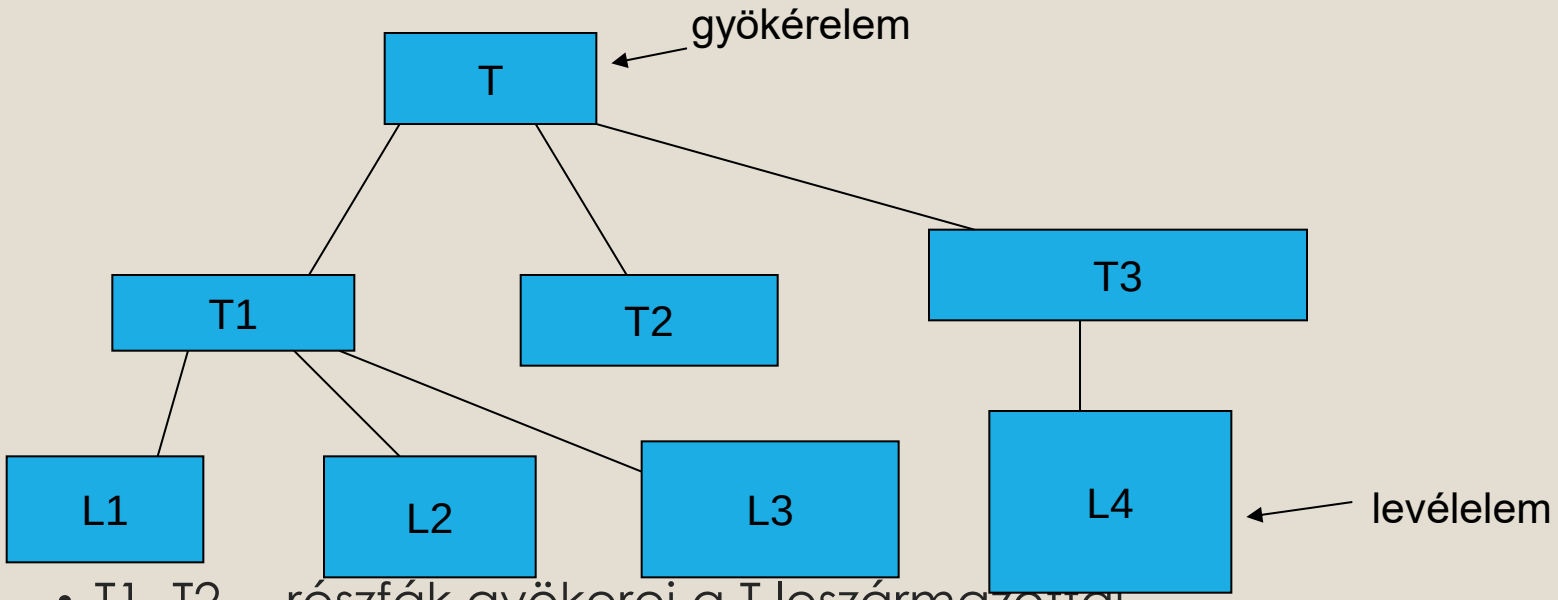
Fa adatszerkezetek

Teszt

- Adja meg a listába illesztés algoritmusát
- Adja meg egy adott elem listába beszúrásának algoritmusát

Fa adatszerkezetek

- A számítógépes algoritmusokban előforduló legfontosabb nem lineáris struktúra.
- Előállítás – rekurzív definíció
  - Azonos típusú rekordokból áll
  - A fa vagy üres
  - Vagy egy kitűntetett csúcshoz (T-hez) kapcsolódó véges sok közös csúcs nélküli T1, T2…Tm fából áll.
  - T a fa gyökere T1,T2…Tm a fa részfái
- (Lineáris lista olyan fa, amelyben minden csúcshoz legfeljebb egy részfa tartozik)



- T1, T2 …részfák gyökerei a T leszármazottai
- T a részfák őse
- L : levélelemek, amelyeknek nincsen leszármazottja
- A csúcs foka: az utódok száma
- Fa foka: a csúcsok fokainak maximuma
- A fa mélysége a gyökérből a levélig vezető maximális úthossz (2)
- Belső úthossz: a belső csúcsok szintjeinek összege
- Külső úthossz: a levelek szintjeinek összege
- Erdő: néhány diszjunkt fából álló halmaz

Hierarchikus szerkezetek leírási módjai. Hol találunk fákat?

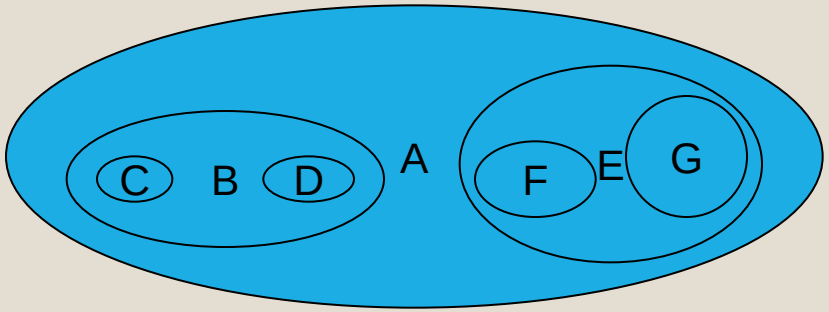
- Gráffal (lásd előző dia)
- Halmazzal

- Zárójeles kifejezéssel
  - (A(B(H)(J))(C(D)))
- Bekezdéses tagolással

A_____
B_____
H_____
J_____
C_____
D_____

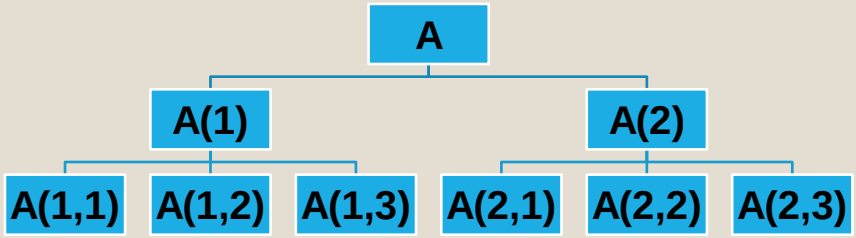
- Tizedes jelöléssel

1 A; 1.1 B; 1.1.1 H; 1.1.2 J; 1.2 C; 1.2.1 D



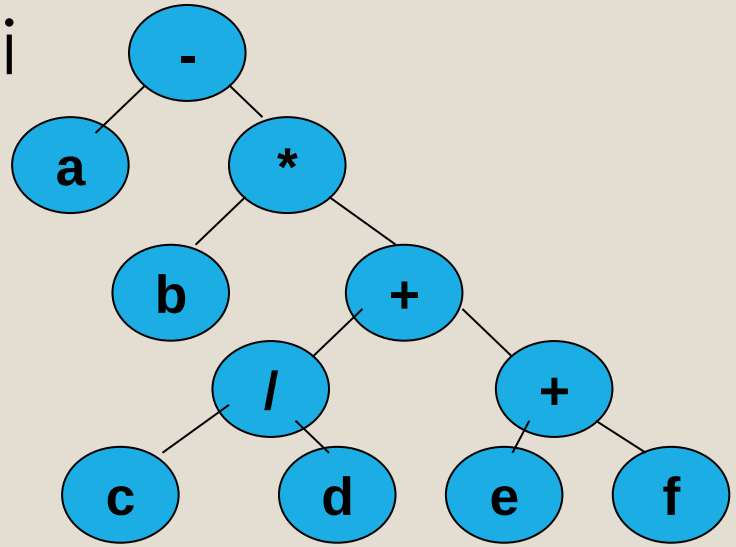
- Minden tömb felfogható a fastruktúra speciális esetének

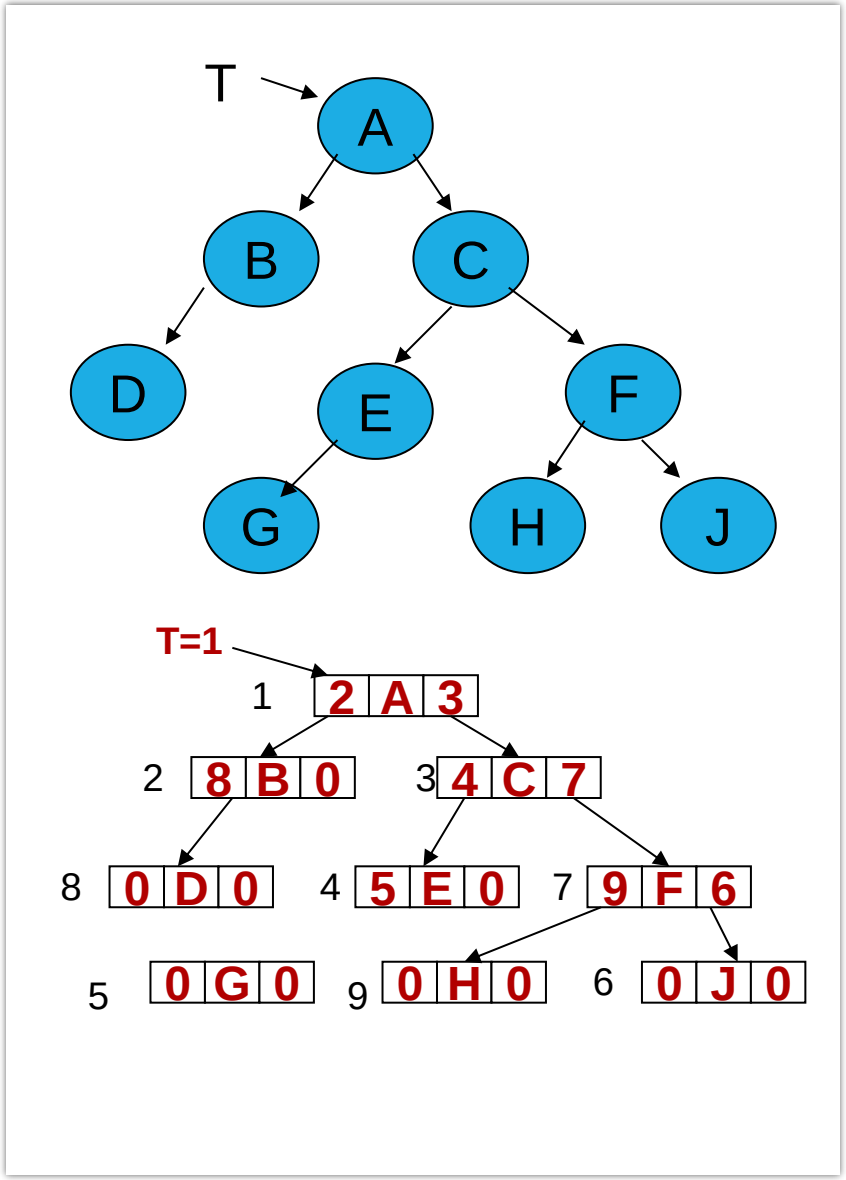
$$\begin{bmatrix} A(1,1) & A(1,2) & A(1,3) \\ A(2,1) & A(2,2) & A(2,3) \end{bmatrix}$$



Bináris fa fogalma

- Csúcsokból álló véges halmaz, amely
  - Vagy üres
  - Vagy egy gyökérből és két (diszjunkt) részfából áll, ezeket bal és jobb részfáknak nevezzük. (Rekurzív definíció!)
- Bináris fákkal ábrázolhatók a matematikai kifejezések
  - Pl. a-b(c/d+e*f)
  - A belső csúcsok az operátorok,a részfák az operandusok

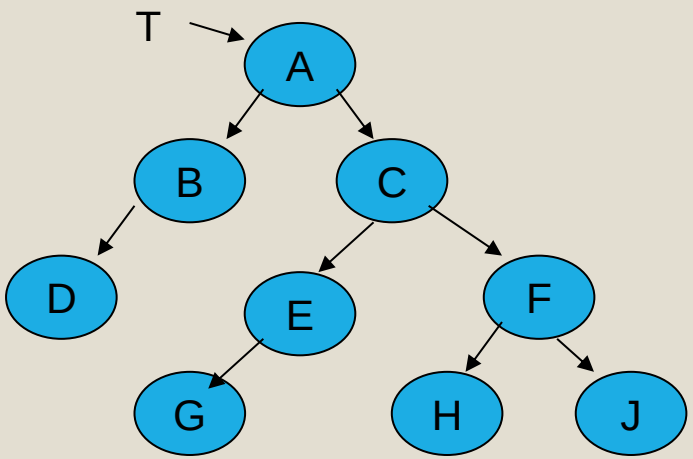




Számítógépes ábrázolás

- Minden csúcsnak egy rekord felel meg, adattartalom, balmutató, jobbmutató mezőkkel
- T mutató a fa gyökerére mutat.
- Ha T=null, akkor a fa üres.

Bináris fa ábrázolása vektorként



Ha a szülő: i
Bal gyerek :2*i+1
Jobb gyerek:2*i+2

A	B	C	D		E	F	G		H	J	
0	1	2	…								

Bináris fák bejárása – Rekurzív algoritmus

- Bejárás a csúcsok módszeres vizsgálatát jelenti, amelynek során minden pontot egyszer és csakis egyszer látogatunk meg.
- A pont meglátogatását nevezzük gyökérvizitnek (Kiírás). A fa teljes bejárása a pontok lineáris sorrendjét adja.
  - Gyökérkezdő – preorder bejárás – közép-bal-jobb
  - Gyökérközepű – inorder bejárás
  - Gyökérvégző – postorder bejárás

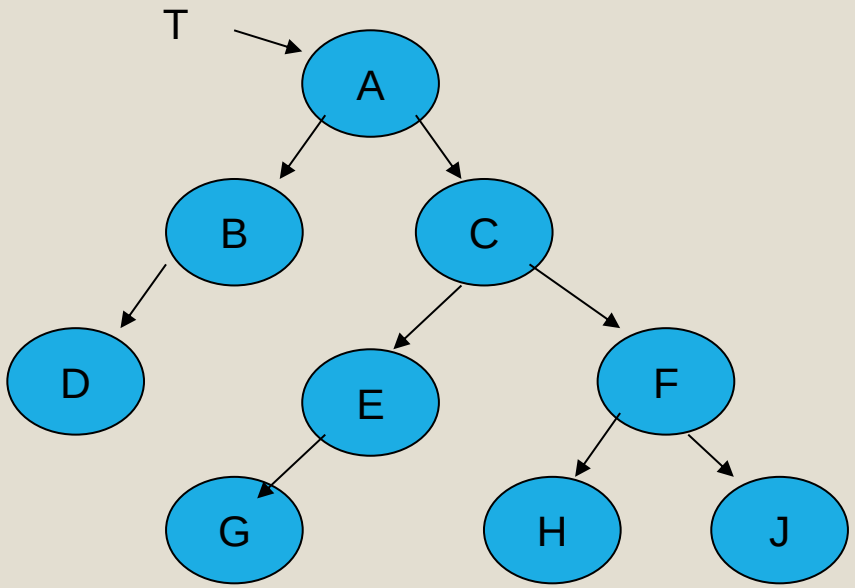
```
Eljárás Preorder(p)
  Ha p!=null akkor
        Feldolgozás(p)
        Preorder(p.bal)
        Preorder(p.jobb)
  Elágazás vége
Eljárás vége
```

- Teljes fa bejárása Preorder(gyökér)
- Gyakorlati alkalmazás fa mentése vagy kiírása

Bejárás algoritmusa (rekurzió!!!)

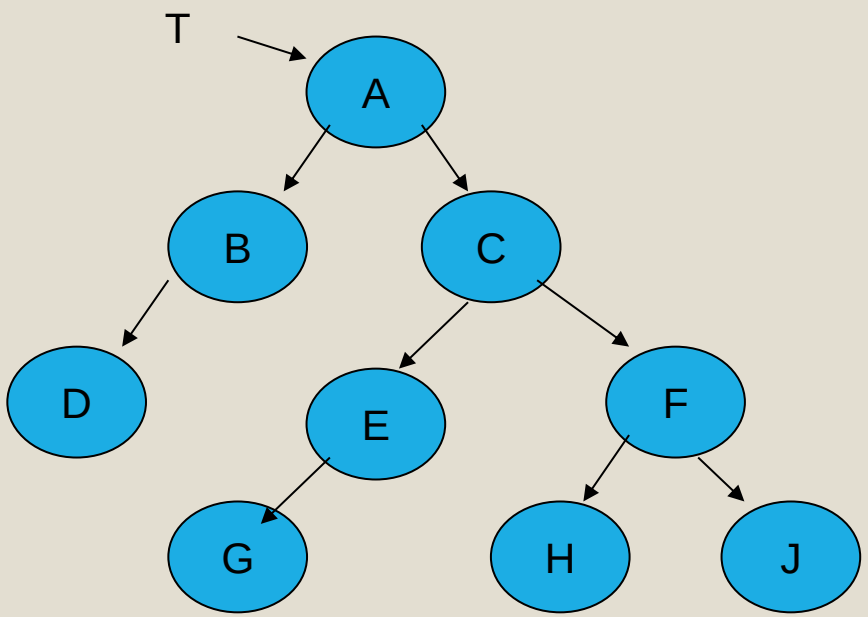
Gyökérkezdő bejárás

- Gyökérvizit
- Bal részfa bejárása gyökérkezdő sorrendben
- Jobb részfa bejárása gyökérkezdő sorrendben
- A B D C E G F H J



Gyökérközepű bejárás

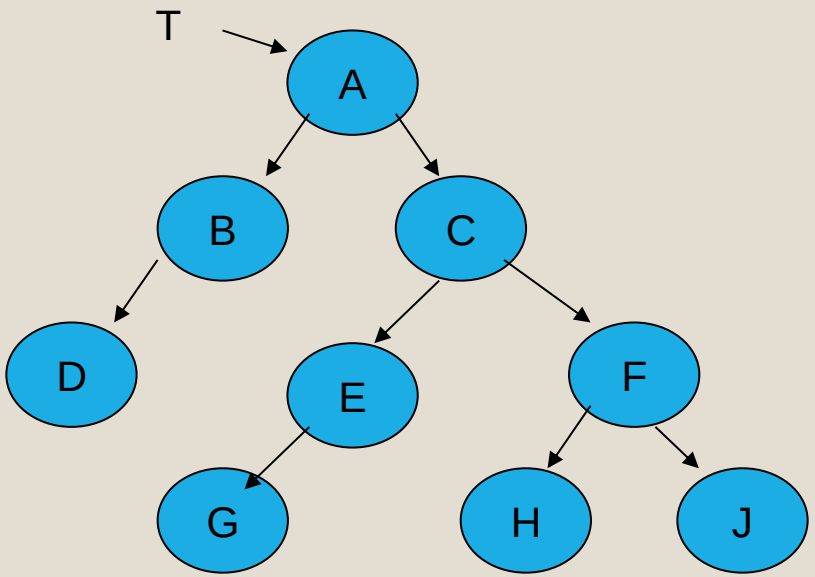
- Bal részfa bejárása gyökérközepű sorrendben
- Gyökérvizit
- Jobb részfa bejárása gyökérközepű sorrendben
- D B A G E C H F J



Gyökérvégző bejárás

- Bal részfa bejárása gyökérvégző sorrendben
- Jobb részfa bejárása gyökérvégző sorrendben
- Gyökérvizit

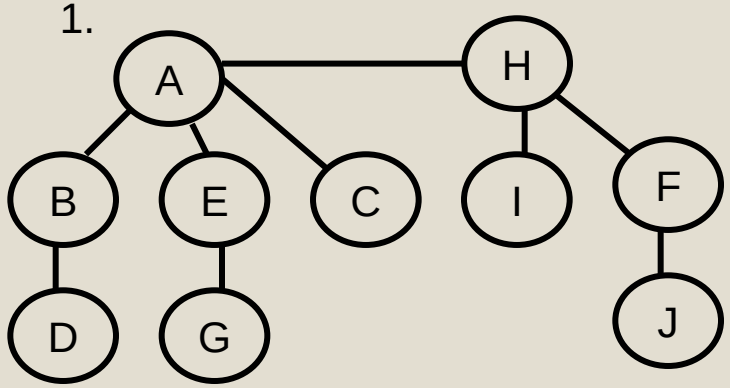
- D B G E H J F C A



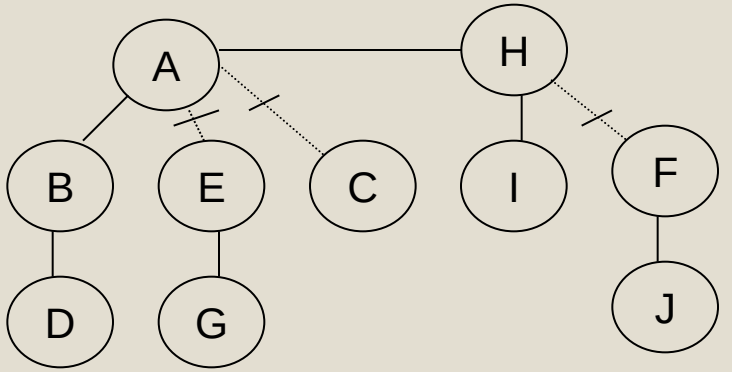
Fák reprezentálása bináris fákkal

- Minden fa (bijektív módon) átalakítható bináris fává

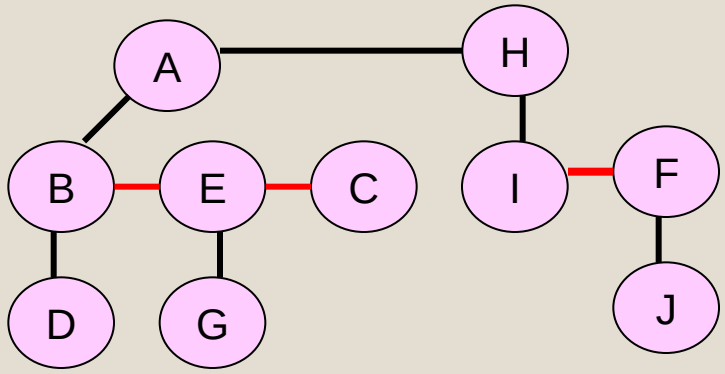
1.



2.



3.



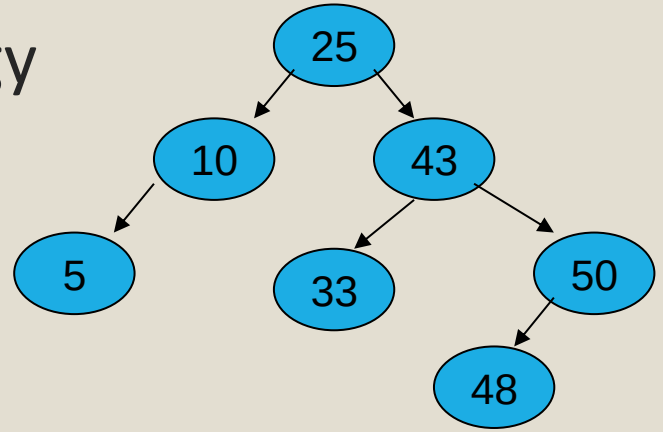
1. Az utódokhoz vezető éleket az első kivételével megszűntetjük
2. Minden család utódait sorba fűzzük

Iteratív fa bejárás

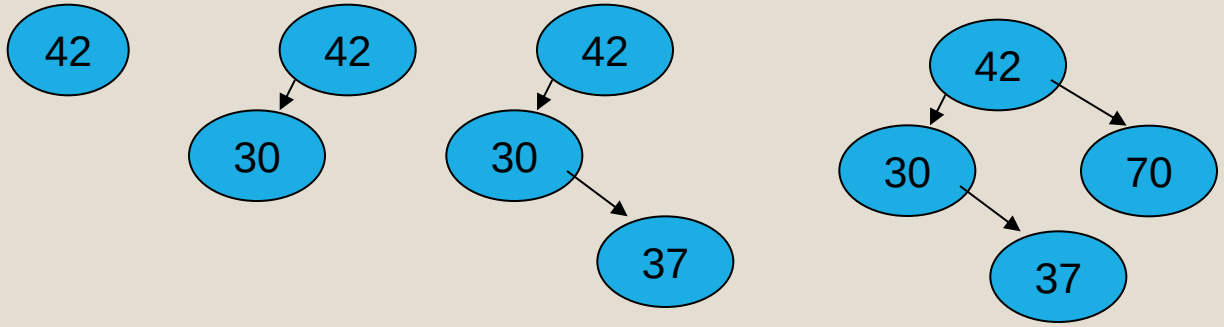
- Sok veremművelettel jár
- Nyilvántartani, és számolni a már bejárt és listázott elemeket

Bináris keresőfák (BST Binary Searching Tree)

- Definíció: Olyan bináris fa, amelyben minden csúcsra igaz, hogy a baloldali részfájában a kulcsok(adattartalom) kisebbek vagy egyenlőek, a jobb oldali részfábanpedig nagyobbak vagy egyenlőek.
- Tetszőleges kulcs elérhető a gyökértől induló keresőútvonalon úgy, hogy minden csúcsból a bal, vagy a jobb részfa felé haladunk, és a döntéshez csak a csúcsot vizsgáljuk.
- Gyökérközepű bejárása?



Keresőfa építése primitív algoritmussal



- 42,30,37,70,50,20,61,57,90,25 elemekből

Beszúrás keresőfába

```
Függvény beszúrás (x,k)
  ha x=null akkor
        elemlétrehozás(x)
        x.kulcs=k x.bal=null x.jobb=null
        return x
  különben
        ha x.kulcs>k akkor
              return beszúrás(x.bal,k)
        különben
              ha x.kulcs<k akkor
              return beszúrás(x.jobb,k)
        return x
Elágazás vége
```

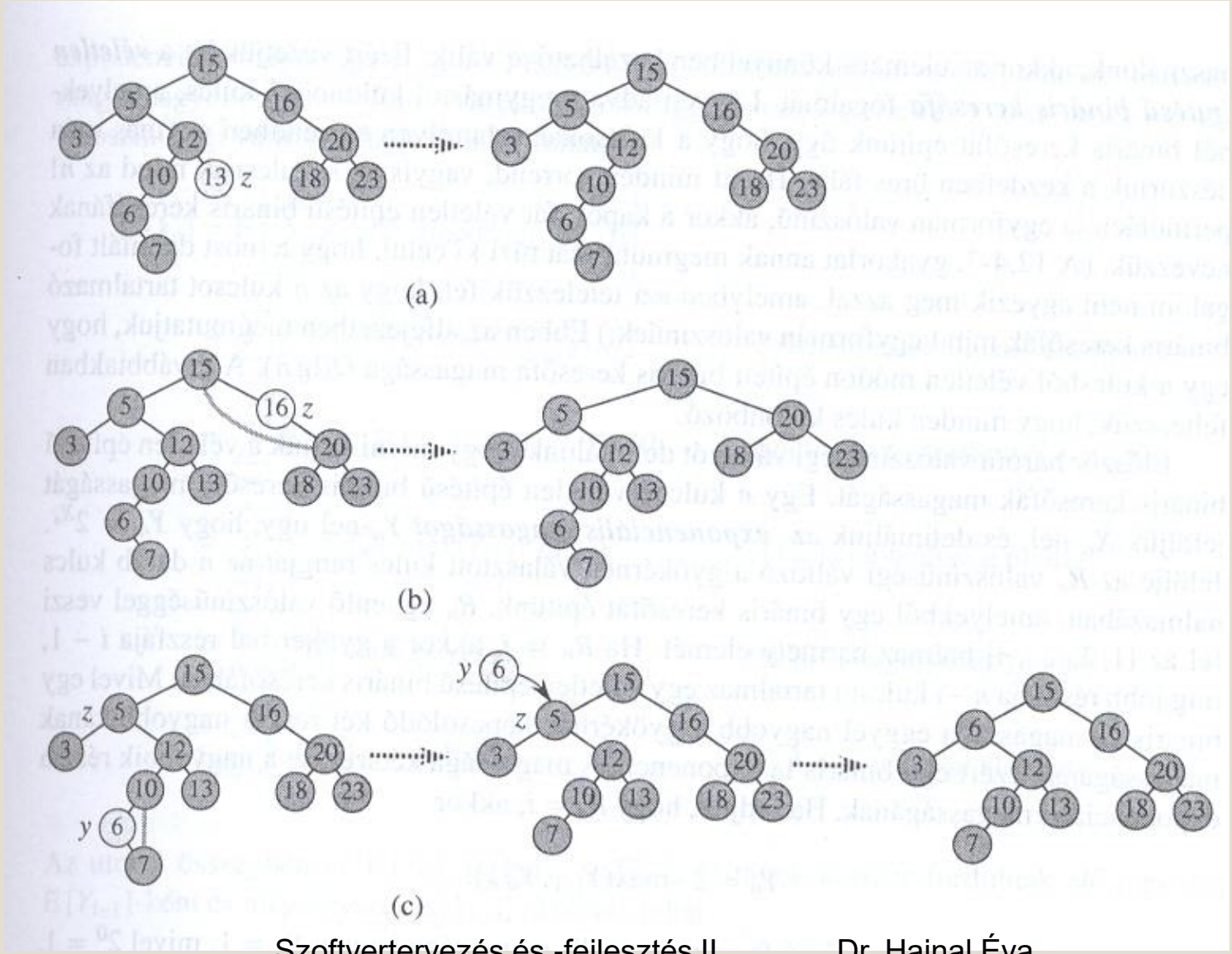
Keresés BST-ben

```
Fában-keres(x,k)        //x a gyökér, k a keresett kulcs
 Ha x=null vagy k=kulcs[x]
          akkor return x
 Ha k<kulcs[x]
     akkor return Fában-keres(bal[x],k)
     különben return Fában-keres(jobb[x],k)
Függvény vége
```

Iteratív keresés

```
Fában-i-keres(x,k)
Ciklus amíg x!=null vagy k!=kulcs[x]
  Ha k<kulcs[x]
     Akkor x=bal[x]
     Különben x=jobb[x]
    Elágazás vége
Ciklus vége
Return x
Függvény vége
```

Törlés bináris fából – 3 eset – rekurzív megoldás

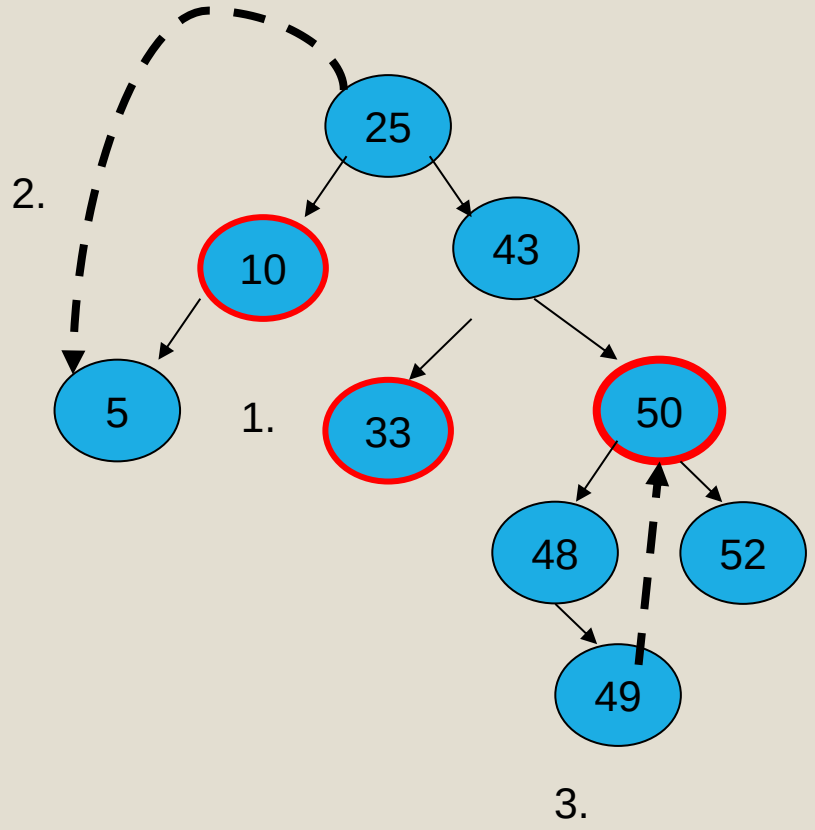


Törlés keresőfából

- Feladat: adott kulcsú elem törlése a keresőfából

Esetek

1. elemnél kisebb→balra le
2. Elemnél nagyobb →jobbra le
3. = akkor
    1. Ha levélelem →törlés (1.eset)
    2. Ha nem levélelem →törlés, és a részfák kapcsolása (2. és 3. eset)



Tökéletesen kiegyensúlyozott fák

- Def. Ha minden csúcsnak a bal és a jobb oldali részfájában lévő csúcsok száma legfeljebb eggyel tér el.

Kiegyensúlyozás

Ha N=0 a fa üres

Ha N>0

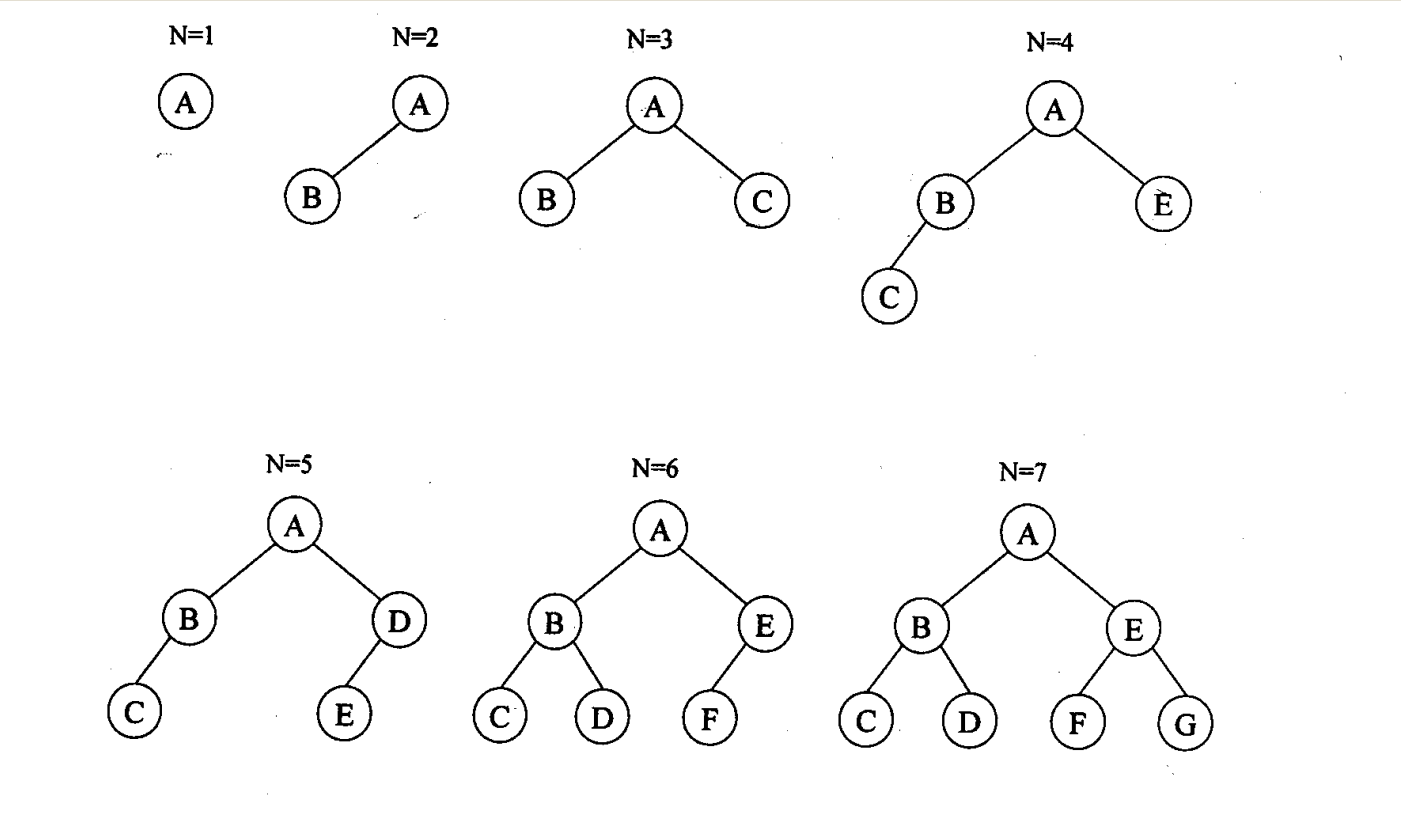
1. Használjunk el egy csúcsot gyökérnek
2. Készítsük el az N/2 csúcsból álló bal részfát
3. Készítsük el az N-(N/2)-1 csúcsból álló jobb részfát

Miért szükséges a keresőfákat kiegyensúlyozni?

Mekkora a keresés időbonyolultsága kiegyensúlyozott és nem kiegyensúlyozott fa esetén?

O(n)
O(logn)

Kiegyensúlyozás



Fák kiegyensúlyozása

- Milyen művelettel lehet egy kiegyensúlyozatlan fát kiegyensúlyozni?

- Forgatás

Kiegyensúlyozott fák

- Def. Ha minden csúcsra legfeljebb egy a különbség a bal és a jobboldali részfa magassága között.
- Ha a keresőfában nagy a magasságbeli különbség a szintek között, akkor a keresés időigénye függ a keresendő elem értékétől.

KÖSZÖNÖM A FIGYELMET!