



# 7. előadás

## Hierarchikus adatszerkezetek

Fa, bináris fa, bejárások

*Adatszerkezetek és algoritmusok* előadás  
2011. március 23.

Kósa Márk és Pánovics János  
Debreceni Egyetem  
Informatikai Kar



## Hierarchikus adatszerkezet

A szekvenciális adatszerkezet általánosítása: minden adatelemének – egyet kivéve – pontosan egy megelőzője, és tetszőleges számú (akár 0) rákövetkezője lehet.

Hierarchikus adatszerkezetek:

- fa
- hierarchikus lista

## Hierarchikus adatszerkezetek

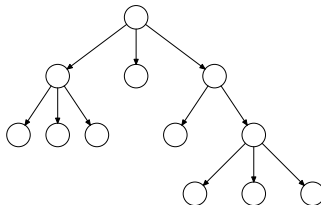
### A fa adatszerkezet

Bináris fa  
Bejárási algoritmusok  
Preorder bejárás  
Inorder bejárás  
Postorder bejárás  
Reprezentáció  
Kifejezésfák



Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

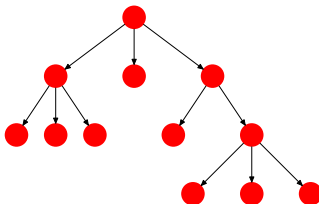
- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- részfa
- szint
- magasság





Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- részfa
- szint
- magasság



## Hierarchikus adatszerkezetek

## A fa adatszerkezet

Bináris fa

## Bejárási algoritmusok

Preorder bejárás

Inorder bejárás:

## Postorder bejárás

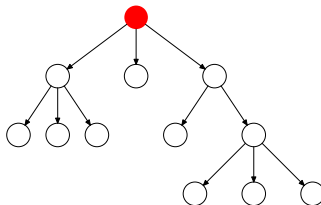
## Reprezentáció

## Kifejezésfák



Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

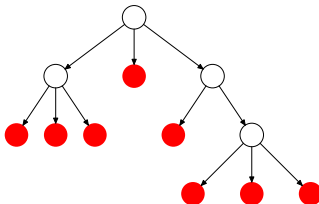
- csúcs, csomópont
- **gyökérelem**
- levélelem
- közbenső elem
- él
- út
- részfa
- szint
- magasság





Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

- csúcs, csomópont
- gyökérelem
- **levélelem**
- közbenső elem
- él
- út
- részfa
- szint
- magasság

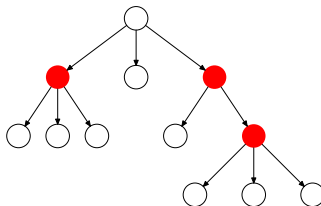


Bináris fa  
Bejárési algoritmusok  
Preorder bejárás  
Inorder bejárás  
Postorder bejárás  
Reprezentáció  
Kifejezősfák



Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

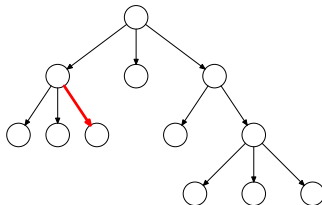
- csúcs, csomópont
- gyökérelem
- levélelem
- **közbenső elem**
- él
- út
- részfa
- szint
- magasság





Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- részfa
- szint
- magasság

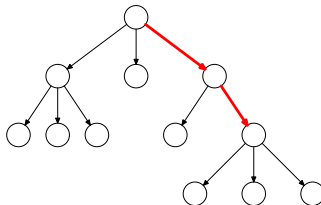






Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

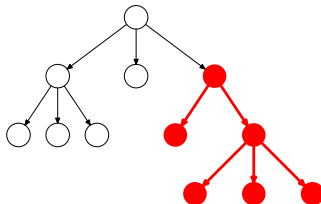
- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- részfa
- szint
- magasság





Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

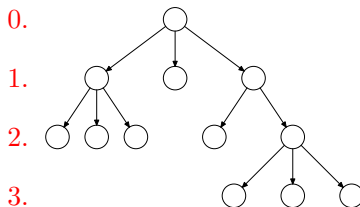
- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- **részfa**
- szint
- magasság





Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

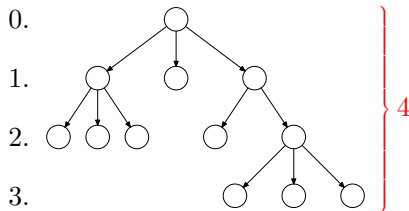
- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- részfa
- **szint**
- magasság





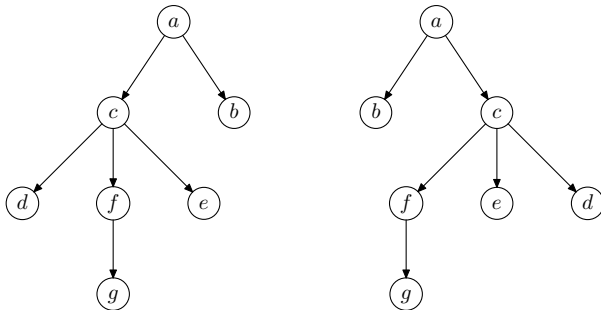
Homogén, dinamikus, hierarchikus adatszerkezet. Fa adatszerkezetekkel kapcsolatos fogalmak:

- csúcs, csomópont
- gyökérelem
- levélelem
- közbenső elem
- él
- út
- részfa
- szint
- **magasság**



## Rendezetlen és rendezett fák

**Rendezetlen** fáknál nem lényeges az ugyanazon csúcsból kiinduló élek sorrendje, **rendezett** fáknál viszont igen.



Az ábrán látható két fa ekvivalens egymással, ha eltekintünk az ugyanazon csúcsokból kiinduló élek sorrendjétől (azaz ha rendezetlen fáknak tekintjük őket).

Mivel az informatikában a rendezett adatszerkezetek játszanak fontos szerepet, a továbbiakban rendezett fákval foglalkozunk.





## Bináris fa

Olyan fa, melyben minden adatelemnek legfeljebb két rákövetkezője van.

## Szigorú értelemben vett bináris fa

Szigorú értelemben vett bináris fáról beszélünk, ha a bináris fában minden adatelemnek 0 vagy 2 rákövetkezője van.

## Rendezett bináris fa

Rendezett bináris fa elemeire értelmezhetők a következő fogalmak:

- bal/jobbs oldali rákövetkező
- bal/jobbs oldali részfa

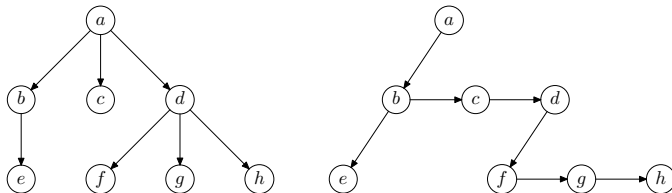
A továbbiakban bináris fa alatt – hacsak mást nem mondunk – rendezett bináris fát értünk.

# Nem bináris fa bináris fává alakítása

Minden nem bináris fa reprezentálható bináris fával.

## Tetszőleges fa binarizálásának algoritmus

- 1 Legyen a bináris fa gyökere a nem bináris fa gyökere.
- 2 A bináris fa egy tetszőleges elemének bal oldali rákövetkezője legyen a nem bináris fa megfelelő elemének bal oldali (első) rákövetkezője.
- 3 A bináris fa egy tetszőleges elemének jobb oldali rákövetkezője legyen a nem bináris fa megfelelő elemének következő (azonos szülőhöz tartozó) testvércsúcsa.





## Bináris fával végezhető műveletek

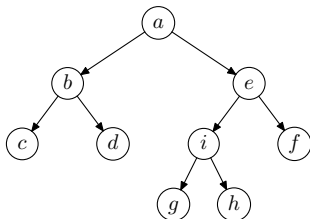
- **Létréhozás**: üres fa.
- **Bővítés**: egy elemmel vagy egy részfával, általában levélelemnél.
- **Törlés**: részfát vagy egy elemet, utóbbi esetben a fát a legtöbb esetben újra kell szervezni (hogy továbbra is fa maradjon).
- **Csere**: megengedett.
- **Rendezés**: nincs.
- **Keresés, elérés és feldolgozás**: a bejárás algoritmusai alapján.
- **Bejárás**: szokás szerint olyan algoritmus, amelynek segítségével a bináris fa elemeit leképezzük egy sorra (preorder, inorder vagy postorder módon).





## Preorder bejárás algoritmusa

- 1 Ha a bejárandó fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.

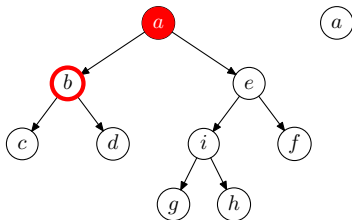


```

graph TD
    a((a)) --> b((b))
    a((a)) --> e((e))
    b((b)) --> c((c))
    b((b)) --> d((d))
    e((e)) --> i((i))
    e((e)) --> f((f))
    i((i)) --> g((g))
    i((i)) --> h((h))
    style a stroke:#f00,stroke-width:2px
  
```

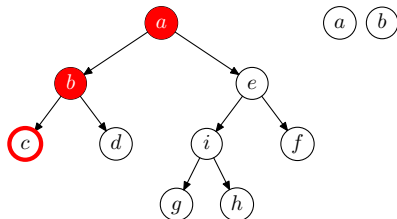
## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemt (más szavakkal: helyezzük a gyökérelemt a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



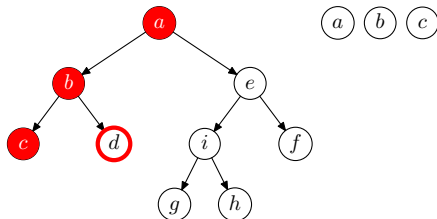
## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemt (más szavakkal: helyezzük a gyökérelemt a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



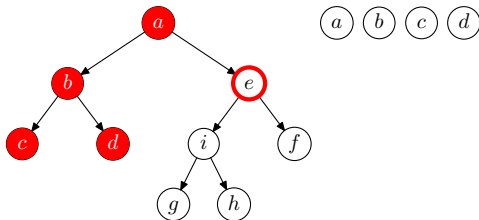
## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



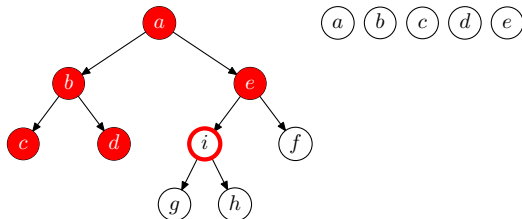
## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemt (más szavakkal: helyezzük a gyökérelemt a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



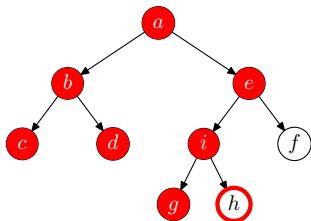
```

graph TD
    a((a)) --> b((b))
    a((a)) --> e((e))
    b((b)) --> c((c))
    b((b)) --> d((d))
    e((e)) --> i((i))
    e((e)) --> f((f))
    i((i)) --> g((g))
    i((i)) --> h((h))
    style a fill:#ff0000
    style b fill:#ff0000
    style c fill:#ff0000
    style d fill:#ff0000
    style e fill:#ff0000
    style f fill:#fff,stroke:#000
    style g fill:#ff0000,stroke:#ff0000,stroke-width:2px
    style h fill:#fff,stroke:#000
    style i fill:#fff,stroke:#000
  
```



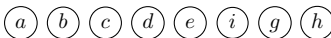
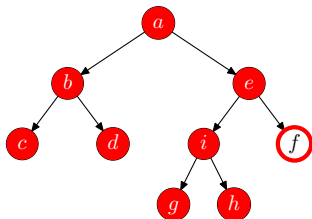
## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



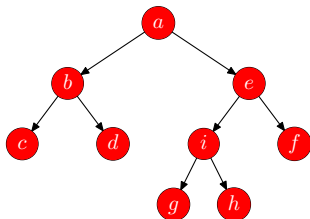
## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



## Preorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Dolgozzuk fel a gyökérelemt (más szavakkal: helyezzük a gyökérelemt a sor végére).
- 3 Járjuk be a gyökérelem bal oldali részfáját preorder módon.
- 4 Járjuk be a gyökérelem jobb oldali részfáját preorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

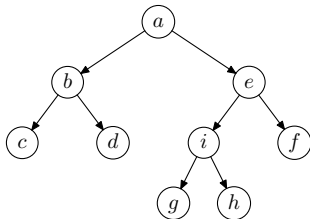
Postorder bejárás

Reprezentáció

Kifejezések

## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

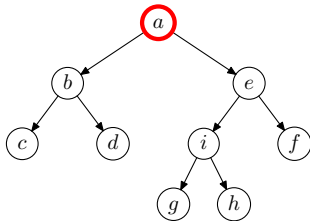
Postorder bejárás

Reprezentáció

Kifejezésfák

## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

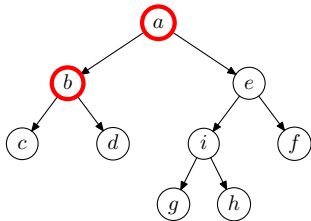
Postorder bejárás

Reprezentáció

Kifejezésfák

## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

Postorder bejárás

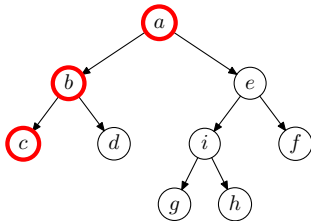
Reprezentáció

Kifejezésfák



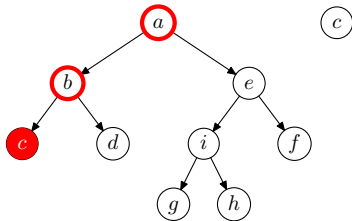
## Inorder bejárás algoritmusa

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

Postorder bejárás

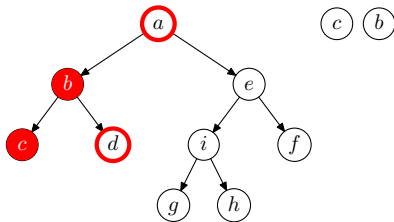
Reprezentáció

Kifejezésfák



## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

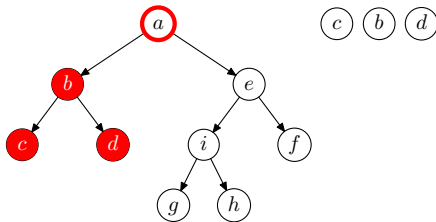
Postorder bejárás

Reprezentáció

Kifejezőesfák

## Inorder bejárás algoritmus

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

Postorder bejárás

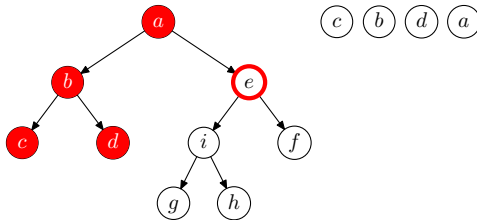
Reprezentáció

Kifejezések



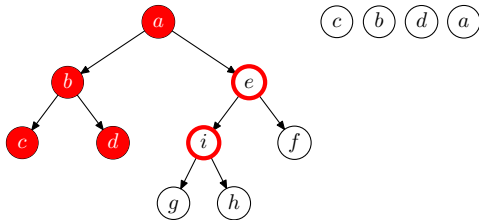
## Inorder bejárás algoritmusa

- 1 Ha a bejárando fá üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



## Inorder bejárás algoritmus

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

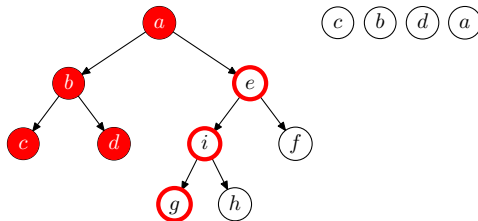
Postorder bejárás

Reprezentáció

Kifejezőfák

## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

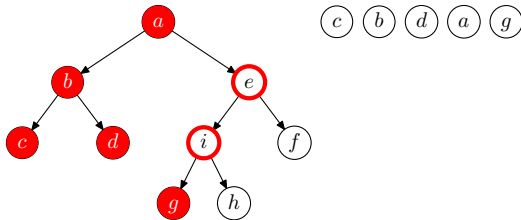
Postorder bejárás

Reprezentáció

Kifejezőfák

## Inorder bejárás algoritmus

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

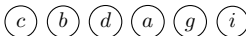
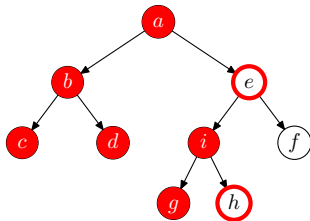
Postorder bejárás

Reprezentáció

Kifejezések

## Inorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

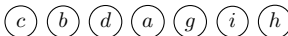
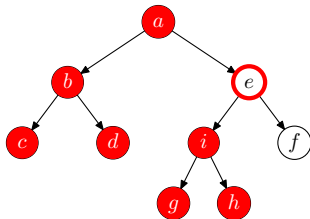
Postorder bejárás

Reprezentáció

Kifejezőesfák

## Inorder bejárás algoritmus

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

Postorder bejárás

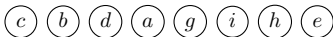
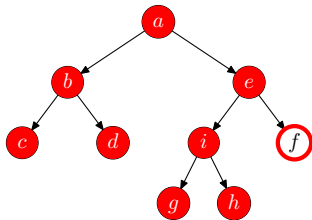
Reprezentáció

Kifejezések



## Inorder bejárás algoritmus

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

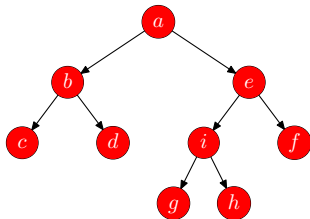
Postorder bejárás

Reprezentáció

Kifejezések

## Inorder bejárás algoritmus

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját inorder módon.
- 3 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).
- 4 Járjuk be a gyökérelem jobb oldali részfáját inorder módon.



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

Postorder bejárás

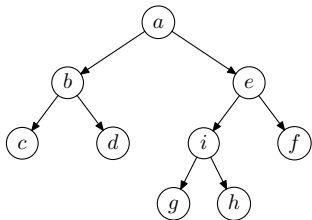
Reprezentáció

Kifejezések



## Postorder bejárás algoritmusa

- 1 Ha a bejárando fá üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Hierarchikus adatszerkezetek

## A fa adatszerkezet

Bináris fa

## Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

### Postorder bejárás

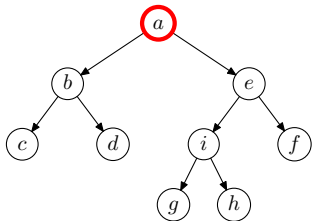
## Reprezentáció

## Kifejezésfák



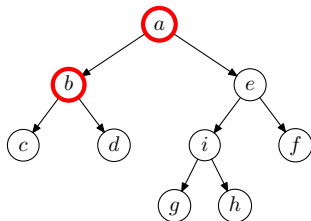
## Postorder bejárás algoritmusa

- 1 Ha a bejárando fá üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



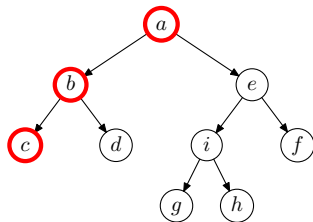
## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



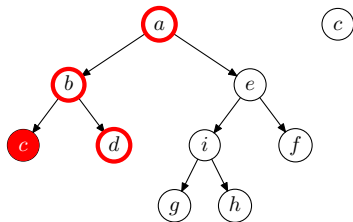
## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



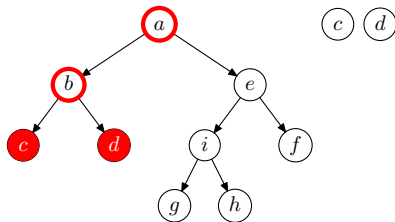
## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Postorder bejárás algoritmusai

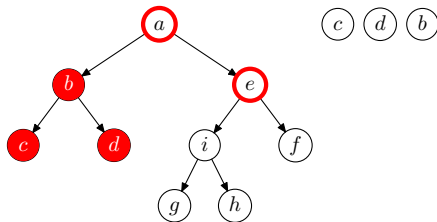
- 1 Ha a bejárandó fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).





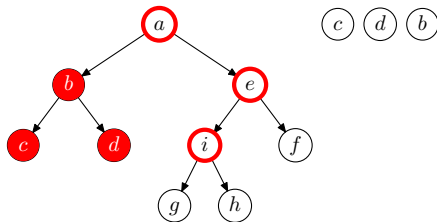
## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Postorder bejárás algoritmusai

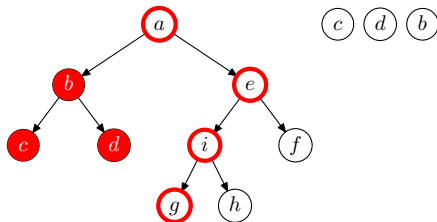
- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).





## Postorder bejárás algoritmusa

- 1 Ha a bejárandó fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Hierarchikus adatszerkezetek

### A fa adatszerkezet

Bináris fa

## Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

## Postorder bejárás

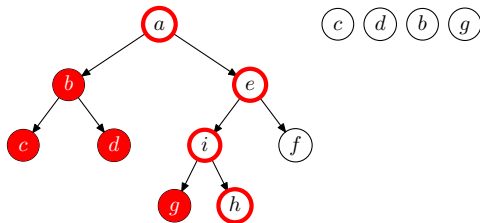
## Reprezentáció

## Kifejezések



## Postorder bejárás algoritmusa

- 1 Ha a bejárandó fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Hierarchikus adatszerkezetek

### A fa adatszerkezet

Bináris fa

## Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

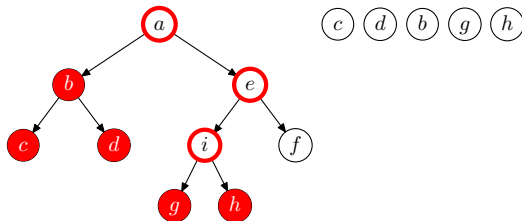
## Postorder bejárás

## Reprezentáció

## Kifejezések

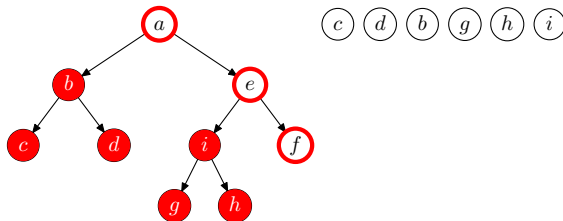
## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

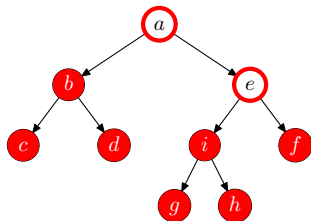
Postorder bejárás

Reprezentáció

Kifejezések

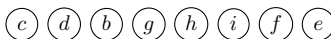
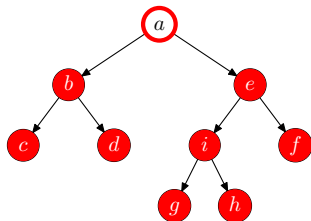
## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).



## Postorder bejárás algoritmusai

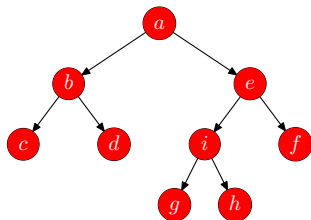
- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).





## Postorder bejárás algoritmusai

- 1 Ha a bejárando fa üres, az algoritmus véget ér.
- 2 Járjuk be a gyökérelem bal oldali részfáját postorder módon.
- 3 Járjuk be a gyökérelem jobb oldali részfáját postorder módon.
- 4 Dolgozzuk fel a gyökérelemet (más szavakkal: helyezzük a gyökérelemet a sor végére).

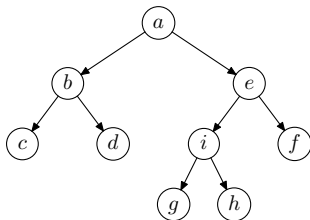


## Bináris fa folytonos reprezentációja

Három vektor segítségével, ahol a vektorok azonos indexű elemei a következő információkat tárolják:

- az ADAT vektorban az adatelem értékét,
- a BAL vektorban a bal oldali rákövetkező vektorbeli indexét,
- a JOBB vektorban a jobb oldali rákövetkező vektorbeli indexét.

Általában a fa gyökérelemét e vektorok első eleme írja le.



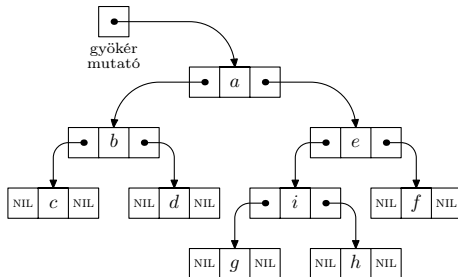
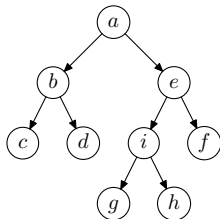
	ADAT	BAL	JOBB
1.	<i>a</i>	2	5
2.	<i>b</i>	3	4
3.	<i>c</i>	0	0
4.	<i>d</i>	0	0
5.	<i>e</i>	6	9
6.	<i>i</i>	7	8
7.	<i>g</i>	0	0
8.	<i>h</i>	0	0
9.	<i>f</i>	0	0



# Bináris fa szétszórt reprezentációja



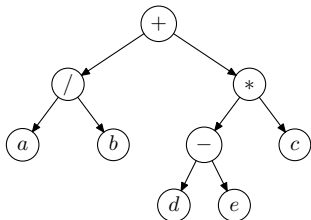
A listaelemek **adatrésze** az adatelem értékét tartalmazza, a **mutatórész** pedig két mutatót: egyet, amely a bal oldali rákövetkezőt leíró listaelemet címzi, és egy másikat, amely a jobb oldali rákövetkezőt leíró listaelemet címzi. A gyökérellemhez (és rajta keresztül az adatszerkezet többi eleméhez) a „**gyökér**” mutató segítségével tudunk hozzáférni.





## Kifejezésfa

A kifejezésfa olyan fa, melyben a levélelemek egy kifejezés operandusait, a nem levél elemek pedig ugyanazon kifejezés operátorait tartalmazzák.



prefix:  $+ / a b * - d e c$

infix:  $a / b + d - e * c$

postfix:  $a b / d e - c * +$

Aszerint, hogy a kifejezésfát – a korábban említettek közül – melyik bejárési algoritmussal járjuk be, kapjuk a kifejezés prefix, infix és postfix alakját.

Hierarchikus  
adatszerkezetek

A fa adatszerkezet

Bináris fa

Bejárási algoritmusok

Preorder bejárás

Inorder bejárás

Postorder bejárás

Reprezentáció

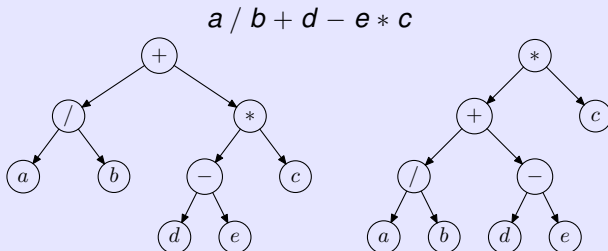
Kifejezésfák

## Kifejezésfák és a bejárások kapcsolata

A prefix és a postfix alak egyértelmű, az infix nem az (de zárójelek használatával azzá tehető).

### Példa

Az alábbi két kifejezésfát inorder módon bejárva ugyanazt az infix kifejezést kapjuk:



Zárójelezést alkalmazva:

$$(a / b) + ((d - e) * c)$$

$$((a / b) + (d - e)) * c$$

