

6. tétel

Adatszerkezetek és algoritmusok vizsga

Frissült: 2013. január 28.

Bináris keresőfák

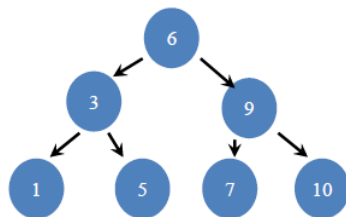
Definíciója

A rendezési fa (vagy keresőfa) olyan bináris fa adatszerkezet, amelynek kialakítása a különböző adatelemek között meglévő rendezési relációt követi.

A fa felépítése olyan, hogy minden csúcásra igaz az, hogy a csúcs értéke **nagyobb**, mint tetszőleges csúcsé a tőle **balra** lévő leszálló ágon és a csúcs értéke **kisebb** minden, a tőle **jobbra** lévő leszálló ágon található csúcs értékénél.

A T fa bármely x csúcsára és $\text{bal}(x)$ bármely y csúcsára és $\text{jobb}(x)$ bármely z csúcsára:

$$y < x < z$$



A rendezési fa az öt tartalmazó elemek beviteli sorrendjét is visszatükrözi.

Inorder bejárással a kulcsok rendezett sorozatát kapjuk.

Inorder bejárás pszeudókódja:

```
Inorder-fa-bejárás(x)
if x ≠ NIL
  then Inorder-fa-bejárás(bal[x])
  print(kulcs[x])
  Inorder-fa-bejárás(jobb[x])
```

Összes érték kiírása: Inorder-fa-bejárás(gyökér[T])

Egy n csúcsú bináris kereső fa bejárása $O(n)$ ideig tart, mivel a kezdőhívás után a fa minden csúcspontja esetében pontosan kétszer (rekurzívan) meghívja önmagát, egyszer a baloldali részfára, egyszer a jobboldali részfára.

Műveletek

Keresés

A T fában keressük a k kulcsú elemet (csúcsot), ha ez létezik, akkor visszaadja az elem címét, egyébként NIL-t.

Rekurzív algoritmus:

```
Fában-keres( $x, k$ )  
  if  $x = \text{NIL}$  or  $k = \text{kulcs}[x]$   
    then return  $x$   
  if  $k < \text{kulcs}[x]$   
    then return Fában-keres( $\text{bal}[x], k$ )  
  else return Fában-keres( $\text{jobb}[x], k$ )
```

Iteratív algoritmus:

```
Fában-iteratívan-keres( $x, k$ )  
  while  $x \neq \text{NIL}$  and  $k \neq \text{kulcs}[x]$   
    do  
      if  $k < \text{kulcs}[x]$   
        then  $x \leftarrow \text{bal}[x]$   
      else  $x \leftarrow \text{jobb}[x]$   
  return  $x$ 
```

Minimum keresés

Tegyük fel, hogy $T \neq \text{NIL}$. Addig követjük a baloldali mutatókat, amíg NIL mutatót nem találunk.

Az iteratív algoritmus pszeudokódja:

```
Fában-minimum ( $T$ )  
   $x \leftarrow \text{gyökér}[T]$   
  while  $\text{bal}[x] \neq \text{NIL}$   
    do  $x \leftarrow \text{bal}[x]$   
  return  $x$ 
```

Helyessége a bináris-kereső-fa tulajdonságból következik.

Lefut $O(h)$ idő alatt, ahol h a fa magassága.

Maximum keresés

Tegyük fel, hogy $T \neq \text{NIL}$. Addig követjük a jobboldali mutatókat, amíg NIL mutatót nem találunk.

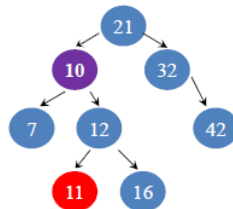
Az iteratív algoritmus pszeudokódja:

```
Fában-maximum ( $T$ )  
   $x \leftarrow \text{gyökér}[T]$   
  while  $\text{jobb}[x] \neq \text{NIL}$   
    do  $x \leftarrow \text{jobb}[x]$   
  return  $x$ 
```

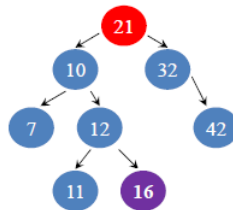
Következő elem

x csúcs rákövetkezőjét adja vissza, ha van, NIL különben.

- Ha van jobb gyereke, jobb részfa minimumát adja vissza.



- Különben felszalad egész akár a gyökérig, míg jobb gyereket talál.



Fában-következő(T, x)

```
if jobb[x]  $\neq$  NIL                                ha van jobb gyereke
    then return Fában-minimum (jobb[x])
y  $\leftarrow$  szülő[x]                                ha nincs
while y  $\neq$  NIL és x = jobb[y]
    do x  $\leftarrow$  y
    y  $\leftarrow$  szülő[x]
return y
```

Beszúrás

- Megkeressük a helyét
- Beláncoljuk

Fába-beszúr (T, p)

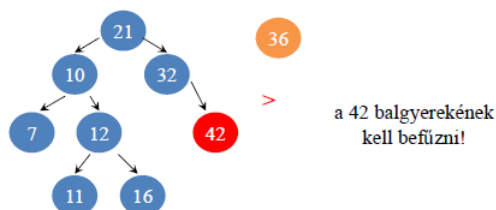
```
y  $\leftarrow$  NIL; x  $\leftarrow$  gyökér[T]
while x  $\neq$  NIL do                                megkeressük a helyét
    y  $\leftarrow$  x
    if kulcs[p] < kulcs[x]
        then x  $\leftarrow$  bal[x]
        else x  $\leftarrow$  jobb[x]
szülő[p]  $\leftarrow$  y
if y = NIL                                        beláncoljuk
```

```

then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
  then bal[y] ← p
  else jobb[y] ← p

```

Pl. beszúrjuk ebbe a fába a 36-os elemet

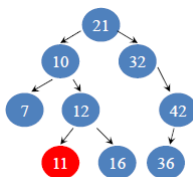


Törlés

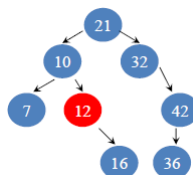
A T bináris keresőfából a p csúcsot töröljük.

Lehetőségek. p -nek

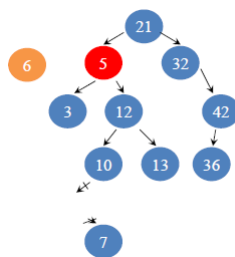
- nincs gyereke: szülőjének mutatóját NIL-re állítjuk



- egy gyereke van: a szülője és a gyermeke között építünk ki kapcsolatot



- két gyereke van: átszervezzük a fát: kivágjuk azt a legközelebbi rákövetkezőjét, aminek nincs balgyereke, így 1., vagy 2. típusú törlés, majd ennek tartalmát beírjuk p -be.



Pszeudókód

Fából-töröl (T,p)

```
if bal[p] = NIL vagy jobb[p] = NIL
  then y ← p
  else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
  then x ← bal[y]
gyerekére
  else x ← jobb[y]
if x ≠ NIL
  then szülő[x] ← szülő[y]
if szülő[y] = NIL
  then gyökér[T] ← x
  else if y = bal[szülő[y]]
    then bal[szülő[y]] ← x
    else jobb[szülő[y]] ← x
if y ≠ p
  then kulcs[p] ← kulcs[y]
return y
```

0 vagy 1 gyerek

2 gyerek

x az y 0 vagy 1

mutat

*ha volt (egy) gyereke,
befűzzük*

ha a gyökeret töröltük

*különben y szülőjének megfelelő
oldali mutatóját*

x-re állítjuk

*ha a log. törlendő ≠ fiz. törlendő
és a további mezők is*