

# ADATSZERKEZETEK ÉS ALGORITMUSOK

Bináris keresőfa

# Rendezési (kereső) fák

- 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 csúcsra 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:

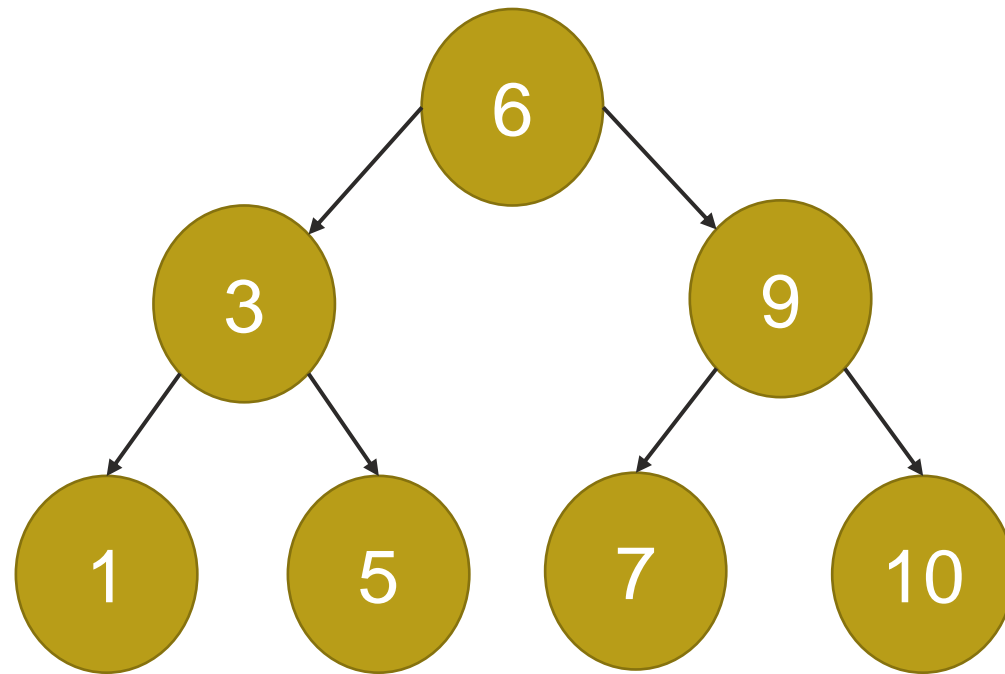
$$y < x < z$$

# Rendezési (kereső) fák

- A rendezési fa az őt tartalmazó elemek beviteli sorrendjét is visszatükrözi.
- Ugyanazokból az elemekből különböző rendezési fák építhetők fel.

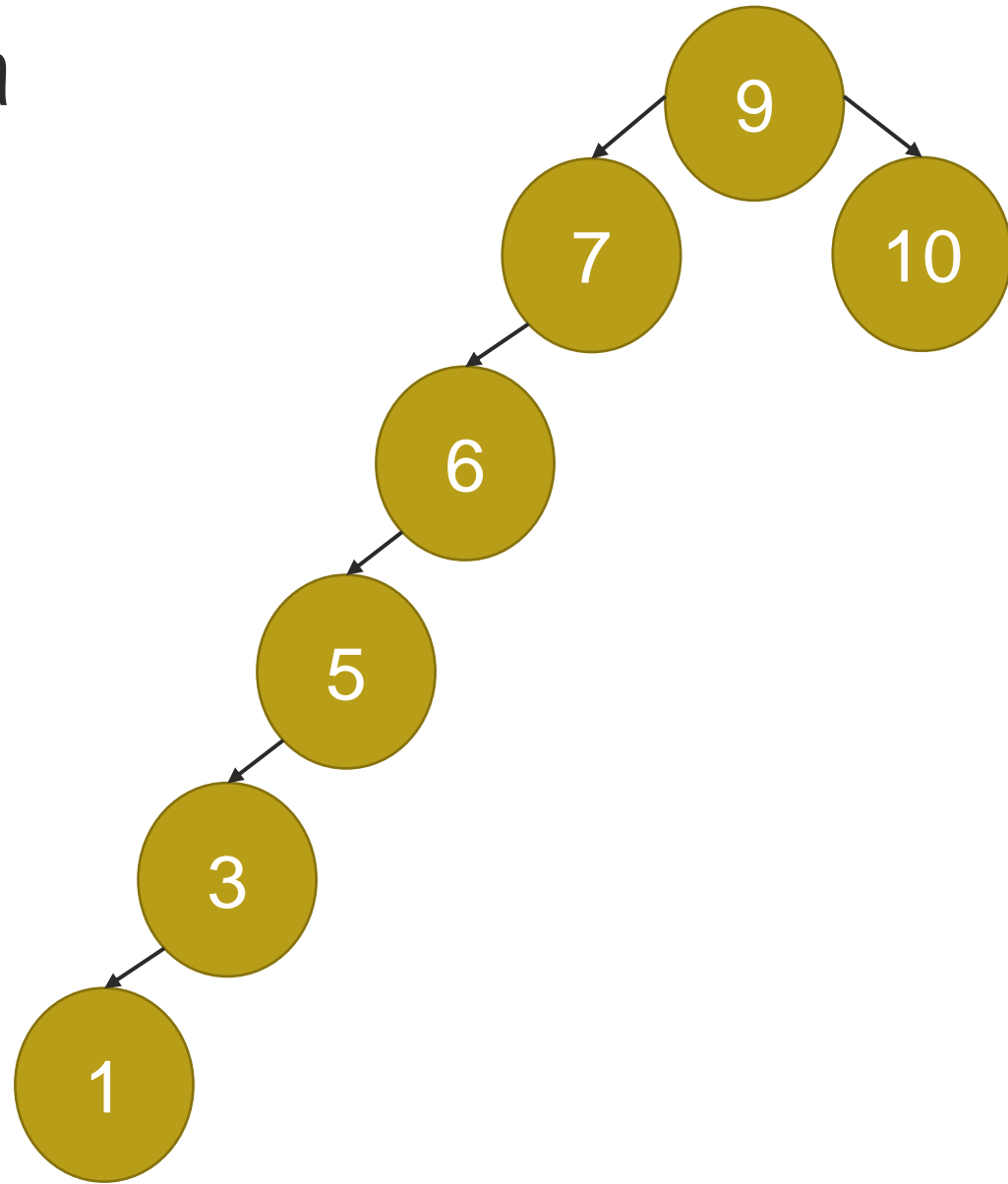
# Elemek beszúrása példa

- 6,3,1,9,7,5,10



# Elemek beszúrása példa

- 9,7,6,5,10,3,1



# Rendezési (kereső) fák

- Fontos tulajdonság
  - inorder bejárással a kulcsok rendezett sorozatát kapjuk
- Az algoritmus pseudokódja:  
Inorder-fa-bejárás(x)  
if  $x \neq \text{NIL}$   
    then Inorder-fa-bejárás(bal[x])  
        print(kulcs[x])  
    Inorder-fa-bejárás(jobb[x])
- Egy T bináris keresőfa összes értékének kiíratásához  
Inorder-fa-bejárás(gyökér[T])

# Rendezési (kereső) fák

- Az algoritmus helyessége a bináris-kereső-fa tulajdonságból indukcióval adódik.
- Egy  $n$  csúcsú bináris kereső fa bejárása  $\mathcal{O}(n)$  ideig tart
  - 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.
- Az algoritmust megadjuk rekurzív és iteratív megoldásban is, ez utóbbi a legtöbb számítógépen hatékonyabb.



# Műveletek

- Keresés

- A rekurzív algoritmus pszeudokódja

Fában-keres(x, k)

```
if x = NIL or k = kulcs[x]
```

```
    then return x
```

```
if k < kulcs[x]
```

```
    then return Fában-keres(bal[x], k)
```

```
    else return Fában-keres(jobb[x], k)
```

# Műveletek

- Keresés:

- Az iteratív algoritmus pszeudokódja

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

# Műveletek

- **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 ← gyökér[T]
while bal[x] ≠ NIL
  do x ← bal[x]
return x
```

- Helyessége a bináris-kereső-fa tulajdonságból következik
- Lefut  $\mathcal{O}(h)$  idő alatt, ahol  $h$  a fa magassága

# Műveletek

- **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)**

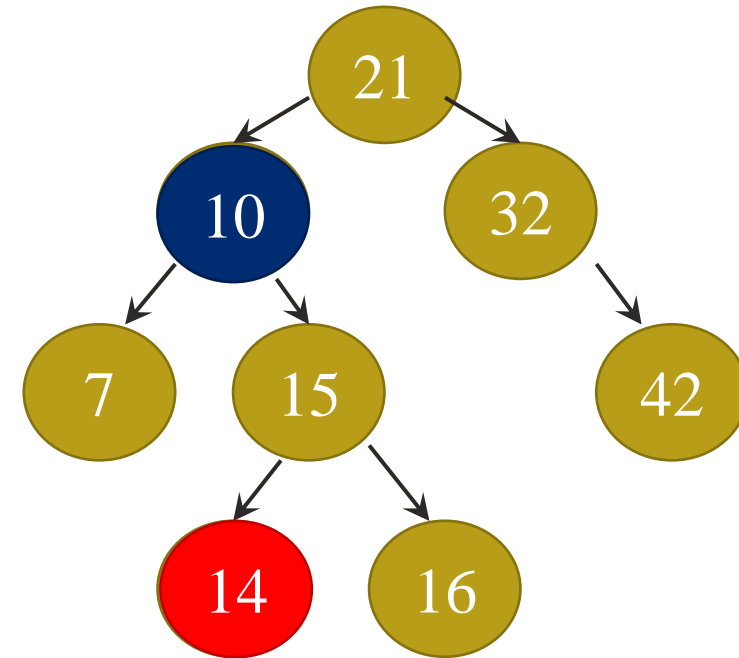
```
x ← gyökér[T]
while jobb[x] ≠ NIL
    do x ← jobb[x]
return x
```

- Helyessége a bináris-kereső-fa tulajdonságból következik
- Lefut  $\mathcal{O}(h)$  idő alatt, ahol  $h$  a fa magassága

# Műveletek

- **Következő elem:** x csúcs rákövetkezőjét adja vissza, ha van, NIL különben

- Több eset lehetséges
- Például a 10 rákövetkezője a 14
  - Létezik a megfelelő jobb részfa



- Algoritmus

```
if jobb[x] ≠ NIL  
    then return Fában-minimum (jobb[x])
```

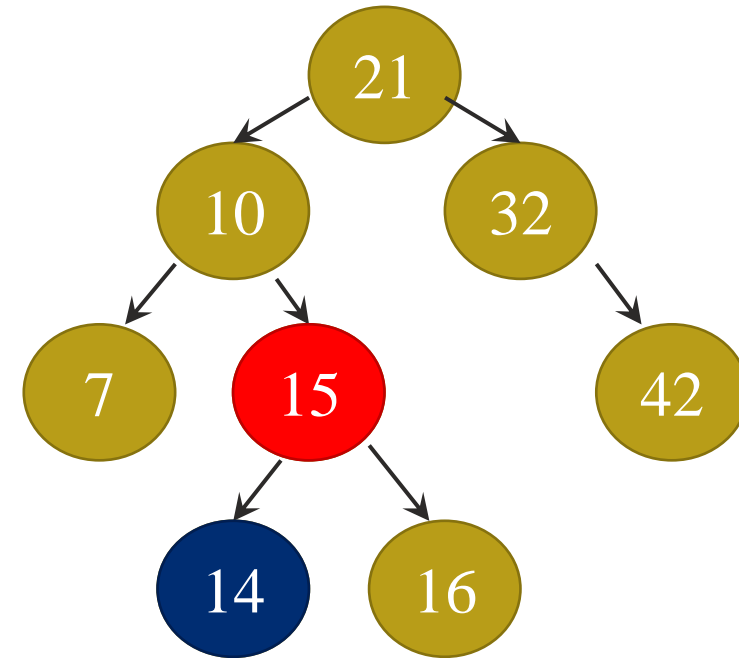
# Műveletek

- **Következő elem:** x csúcs rákövetkezőjét adja vissza, ha van, NIL különben

- A 14 rákövetkezője a 15
  - Nem létezik a jobb részfa
    - Felfelé kell keresni

- Algoritmus

```
y ← szülő[x]  
while y ≠ NIL és x = jobb[y] do  
    x ← y  
    y ← szülő[x]  
return y
```



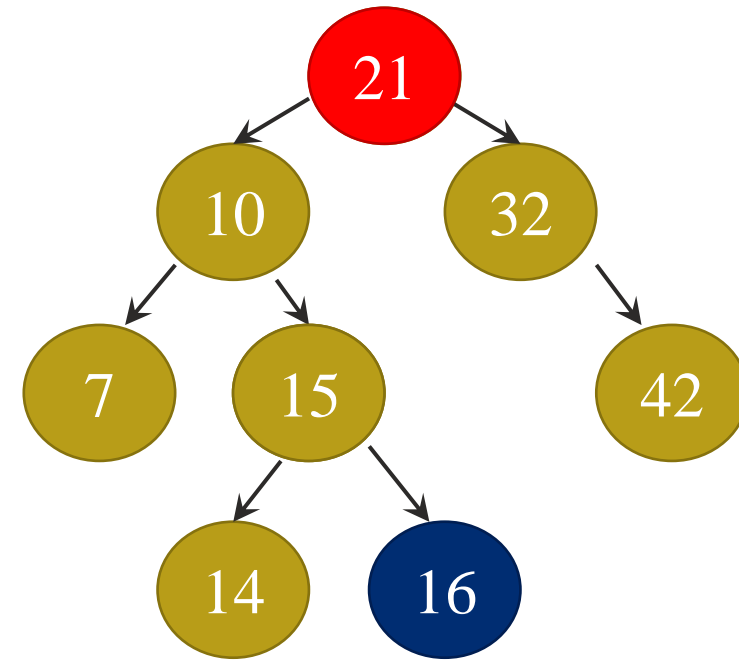
# Műveletek

- **Következő elem:** x csúcs rákövetkezőjét adja vissza, ha van, NIL különben

- A 16 rákövetkezője a 21
  - Nem biztos, hogy mindig a gyökér!

- Algoritmus

```
y ← szülő[x]  
while y ≠ NIL és x = jobb[y] do  
    x ← y  
    y ← szülő[x]  
return y
```



# Műveletek

- Következő elem: x csúcs rákövetkezőjét adja vissza, ha van, NIL különben
- Teljes algoritmus

**Fában-következő(T, x)**

if jobb[x]  $\neq$  NIL

    then return Fában-minimum (jobb[x])

y  $\leftarrow$  szülő[x]

while y  $\neq$  NIL és x = jobb[y] do

    x  $\leftarrow$  y

    y  $\leftarrow$  szülő[x]

return y



# Műveletek

- **Fában-következő**( $T, x$ ) futási ideje  $h$  magasságú fák esetén  $\mathcal{O}(h)$ .
- Megelőző elem:  $x$  csúcs megelőzőjét adja vissza, ha van, NIL különben.
  - **Fában-megelőző**( $T, x$ )
  - Házi feladat

# Műveletek

- Tétel: A dinamikus halmazokra vonatkozó Keres, Minimum, Maximum, Következő és Előző műveletek  $h$  magasságú bináris keresőfában  $\mathcal{O}(h)$  idő alatt végezhetők el
  - Bizonyítás: az előzőekből következik

# Műveletek

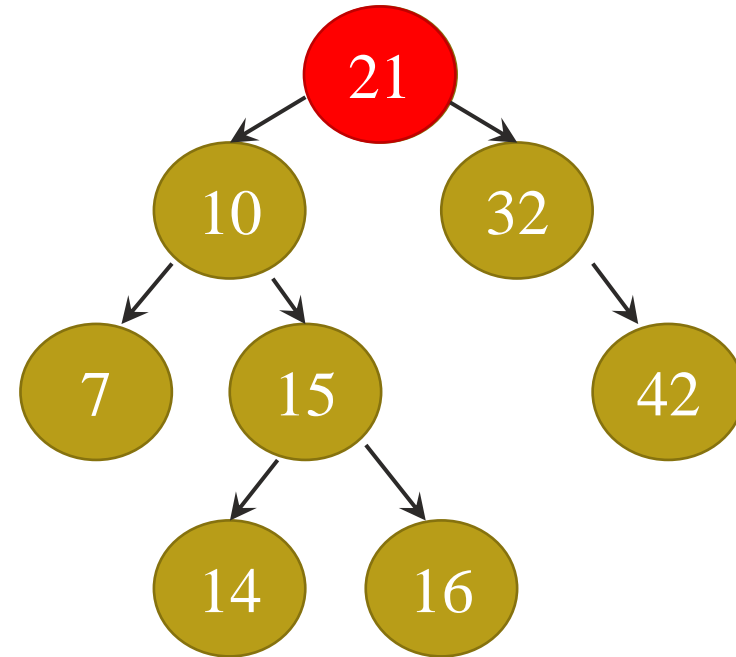
- **Beszúrás**

- A T bináris keresőfába a p csúcsot szúrjuk be.
- Kezdetben:
  - `kulcs[p]=k`
  - `bal[p] = NIL`
  - `jobb[p] = NIL`
  - `szülő[p] = NIL`
- Feltételezzük, hogy a fában még nincs k kulcsú csúcs!
  - Otthoni feladat megnézni, hogyan változik az algoritmus, ha ez a feltételezés nem igaz

# Műveletek

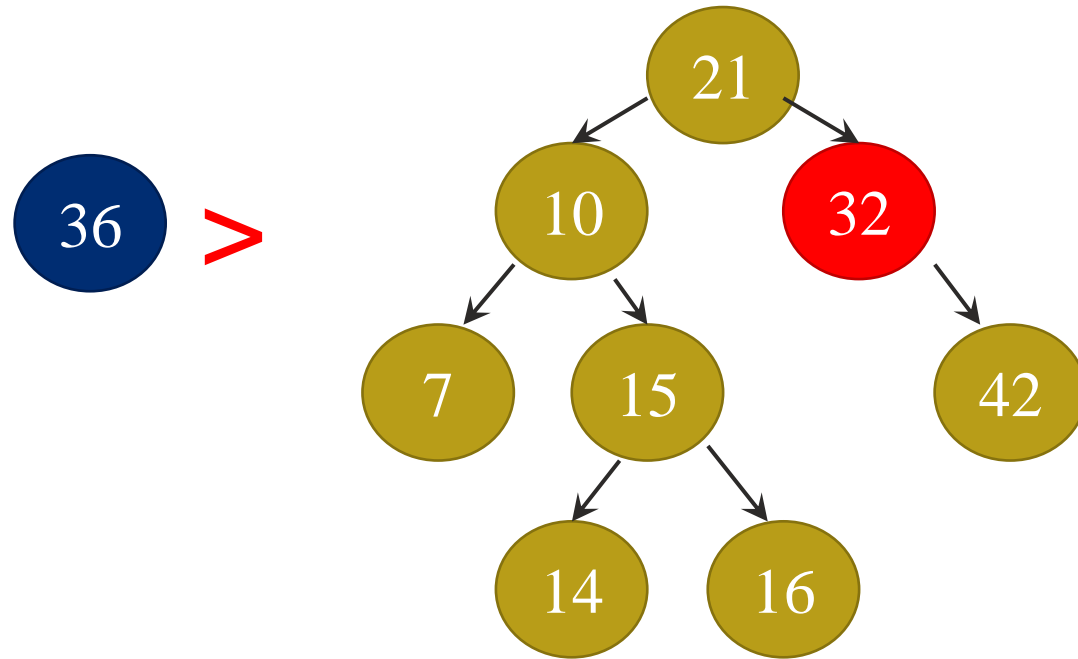
- **Fába beszúr:** szúrjuk be például a 36-t!
  - 1. megkeressük a helyét

36 >



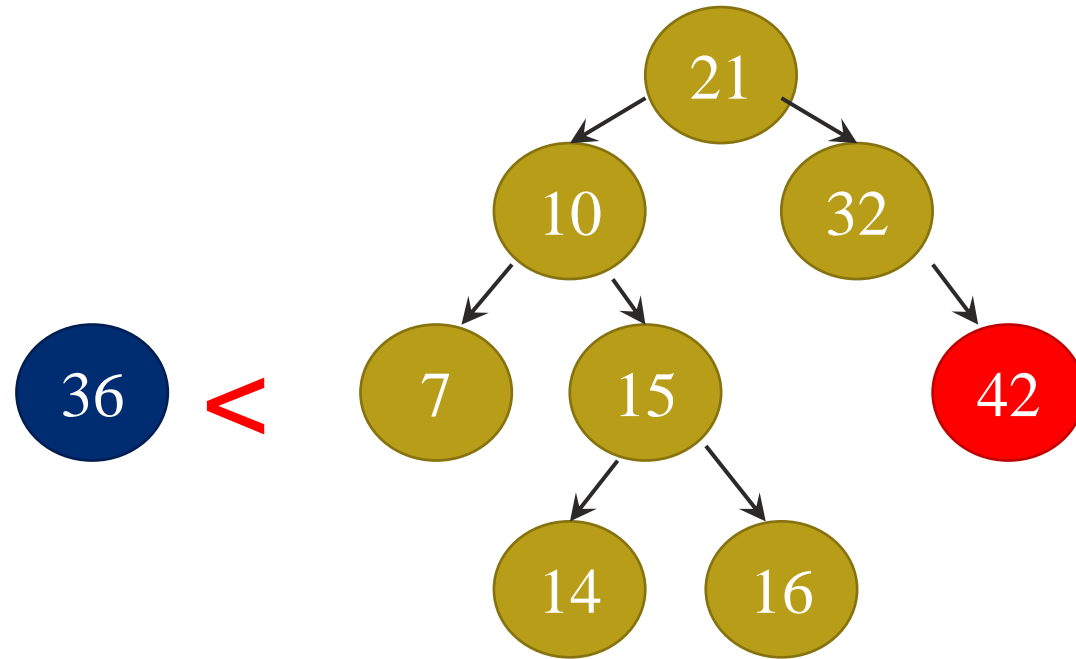
# Műveletek

- **Fába beszúr:** szúrjuk be például a 36-t!
  - 1. megkeressük a helyét



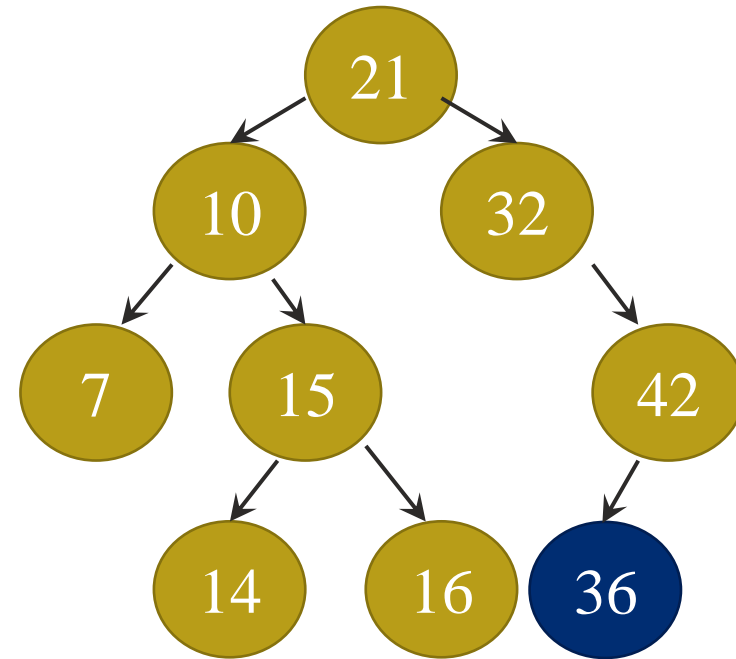
# Műveletek

- **Fába beszúr:** szúrjuk be például a 36-t!
  - 1. megkeressük a helyét



# Műveletek

- **Fába beszúr:** szúrjuk be például a 36-t!
  - 1. megkeressük a helyét
  - 2. beláncoljuk



# Műveletek

- Algoritmus

Fába-beszúr (T,p)

y ← NIL; x ← gyökér[T]

while x ≠ NIL

do y ← x

if kulcs[p] < kulcs[x]

then x ← bal[x]

else x ← jobb[x]

szülő[p] ← y

if y = NIL

then gyökér[T] ← p

else if kulcs[p] < kulcs[y]

then bal[y] ← p

else jobb[y] ← p



# Műveletek

- Törlés:

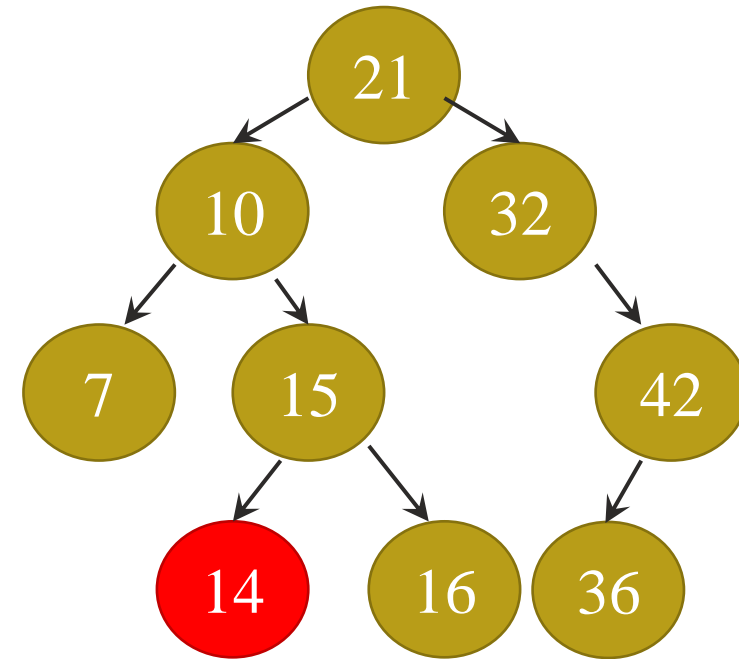
- A T bináris keresőfából a p csúcsot töröljük

- Lehetőségek:

1. p-nek még **nincs gyereke**: szülőjének mutatóját NIL-re állítjuk
2. p-nek **egy gyereke van**: a szülője és a gyermeke között építünk ki kapcsolatot
3. p-nek **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

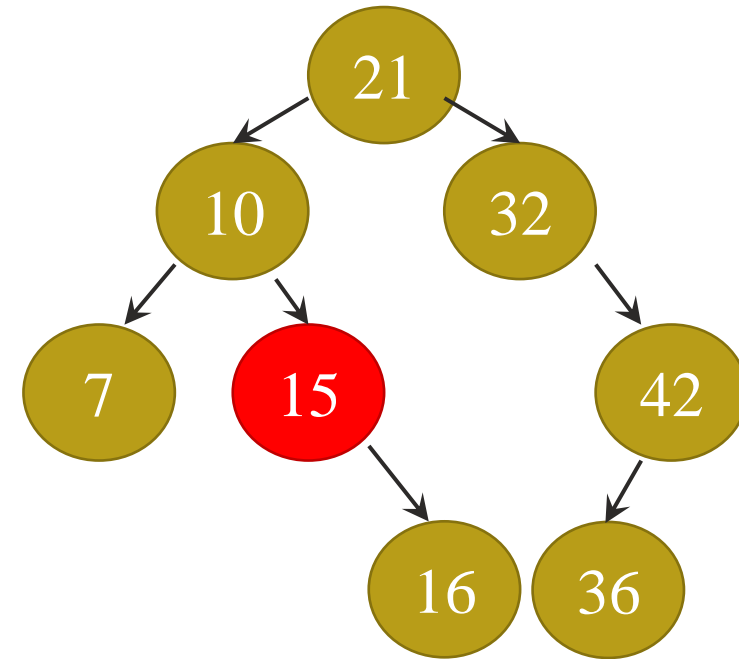
# Műveletek

- **Törlés:** töröljük ki például a 14-t!
  - Ez a legegyszerűbb eset, alkalmazzuk az 1. szabály szerinti teendőket



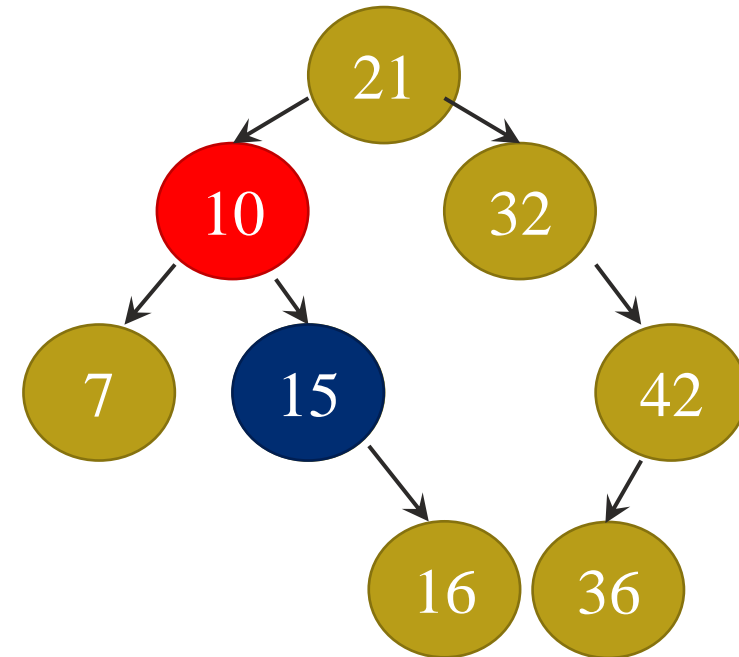
# Műveletek

- **Törlés:** töröljük ki például a 15-t!
  - Ebben az esetben egy gyereke van a törlendőnek, tehát a 2. szabályt alkalmazzuk



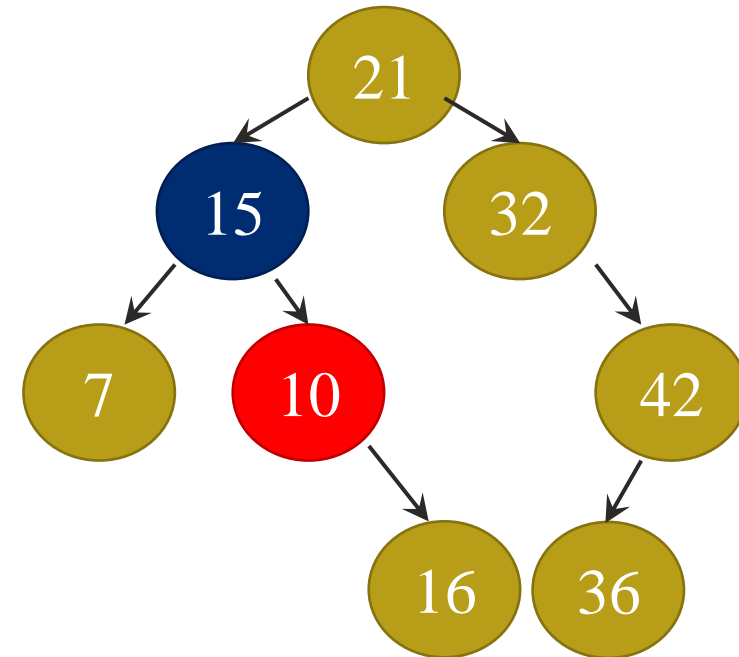
# Műveletek

- **Törlés:** töröljük ki például a 10-et!
  - Ebben az esetben két gyereke van a törlendőnek, tehát a 3. szabályt alkalmazzuk
    - A megfelelő, rákövetkező elem a 15.
    - Ennek nincsen balgyereke
      - Ha lenne nem az lenne a rákövetkező
    - Előfordulhat, hogy jobbgyereke sincs



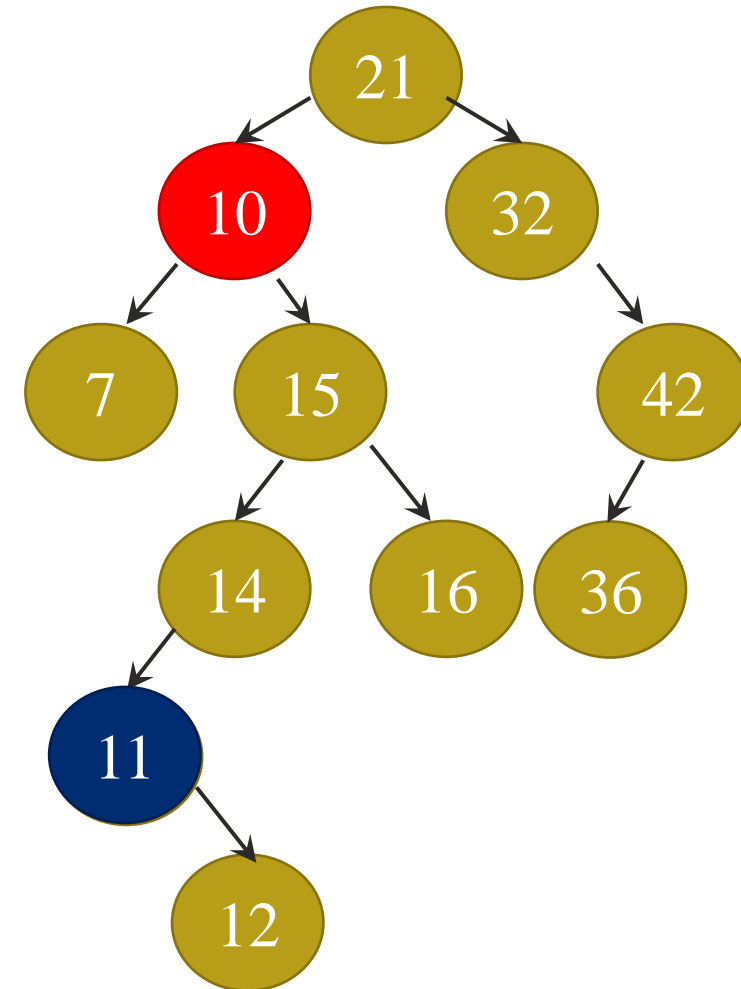
# Műveletek

- **Törlés:** töröljük ki például a 10-et!
  - Ebben az esetben két gyereke van a törlendőnek, tehát a 3. szabályt alkalmazzuk
    - A megfelelő, rákövetkező elem a 15.
    - Ennek nincsen balgyereke
      - Ha lenne nem az lenne a rákövetkező
    - Előfordulhat, hogy jobbgyereke sincs
  - Helyet cserél a törlendő és a rákövetkező
    - Majd végrehajtjuk a törlést az eddigiek szerint



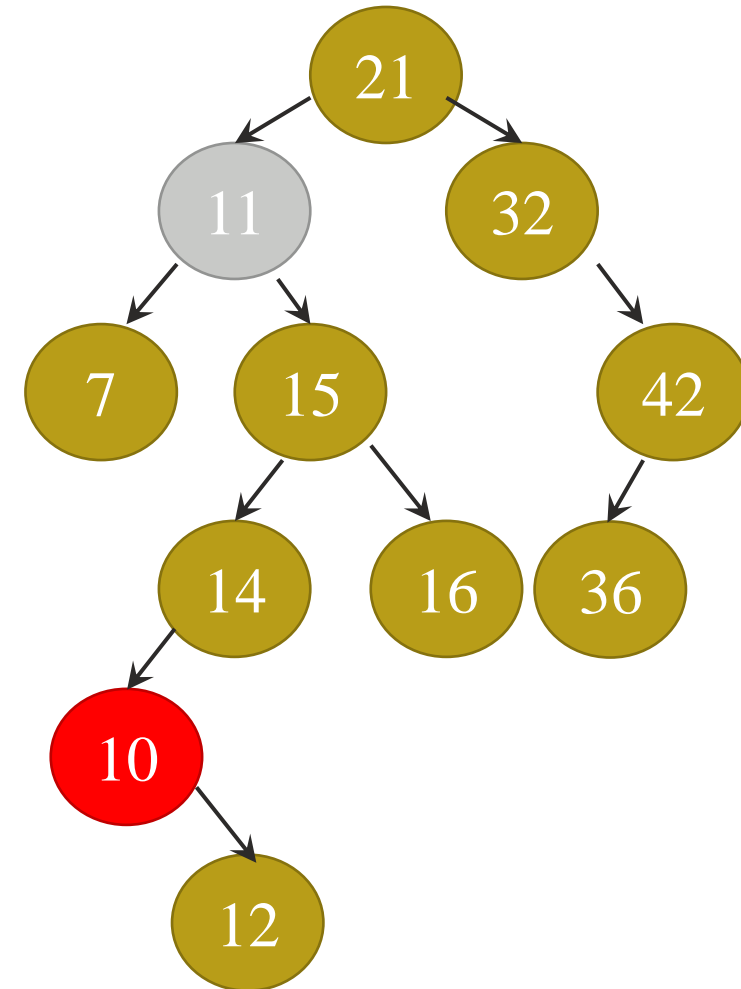
# Műveletek

- **Törlés:** töröljük ki például a 10-et!
  - Ugyanez komplikáltabb példán



# Műveletek

- **Törlés:** töröljük ki például a 10-et!
  - Ugyanez komplikáltabb példán



# Műveletek

- Feltesszük, hogy  $p$  a  $T$ -ben létezik

- **Fából-töröl ( $T, p$ )**

```
if bal[p] = NIL vagy jobb[p] = NIL
  then  $y \leftarrow p$ 
  else  $y \leftarrow$  Fában-következő( $T, p$ )
if bal[y]  $\neq$  NIL
  then  $x \leftarrow$  bal[y]
  else  $x \leftarrow$  jobb[y]
if  $x \neq$  NIL
  then szülő[x]  $\leftarrow$  szülő[y]
if szülő[y] = NIL
  then gyökér[T]  $\leftarrow$  x
  else if  $y =$  bal[szülő[y]]
    then bal[szülő[y]]  $\leftarrow$  x
    else jobb[szülő[y]]  $\leftarrow$  x
if  $y \neq p$ 
  then kulcs[p]  $\leftarrow$  kulcs[y]
return y
```

-- 0, vagy 1 gyerek

-- 2 gyerek

--  $x$  az  $y$  0, vagy 1 gyerekére mutat

-- ha volt gyereke, akkor befűzi az új szülőhöz

-- ha a gyökeret töröltük akkor be kell állítani az újat  
-- különben a szülőnek megfelelő oldalhoz tartozó  
mutatót kell az  $x$ -re állítani

-- amennyiben a ténylegesen törlendő csúcs nem  
azonos azzal, amit kiláncolunk át kellírni az  
adatot is



# Keresés hatékonysága

Következő téma

Előtte további animáció

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 10

Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 10



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

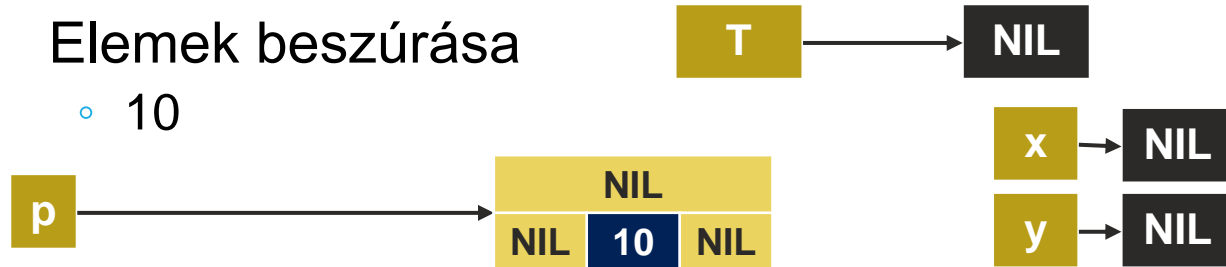
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 10



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

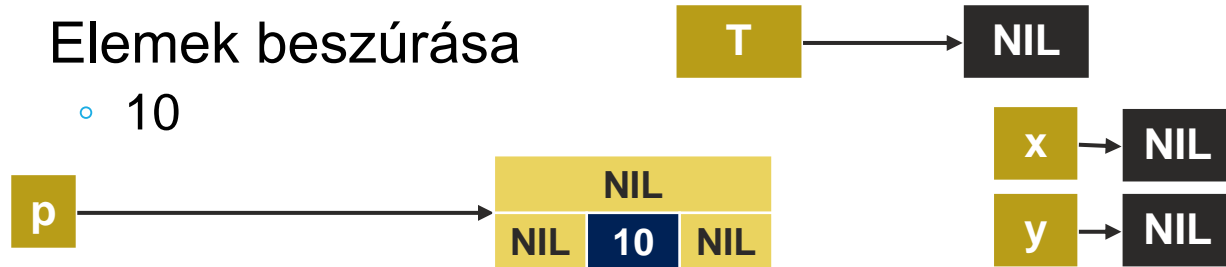
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 10



Fába-beszúr (T,p)

`y ← NIL; x ← gyökér[T]`

**while** `x ≠ NIL`

do `y ← x`

if `kulcs[p] < kulcs[x]`

then `x ← bal[x]`

else `x ← jobb[x]`

`szülő[p] ← y`

if `y = NIL`

then `gyökér[T] ← p`

else if `kulcs[p] < kulcs[y]`

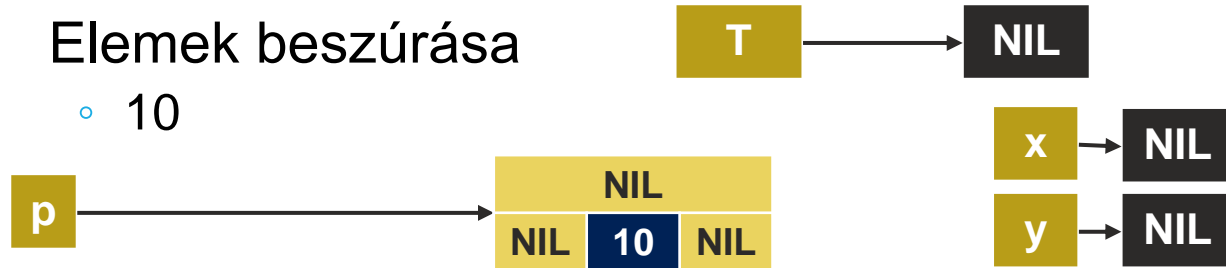
then `bal[y] ← p`

else `jobb[y] ← p`

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 10



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

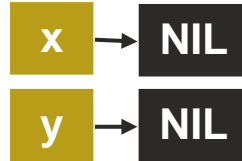
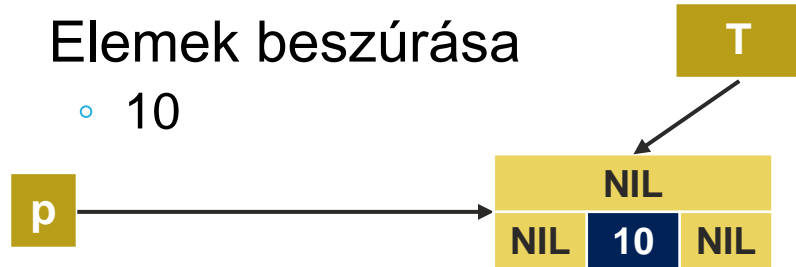
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 10



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

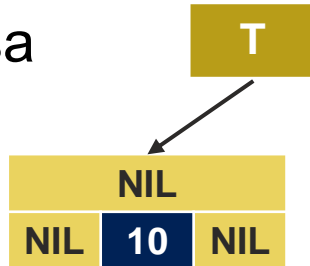
else if  $\text{kulcs}[p] < \text{kulcs}[y]$

then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

then  $\text{bal}[y] \leftarrow p$

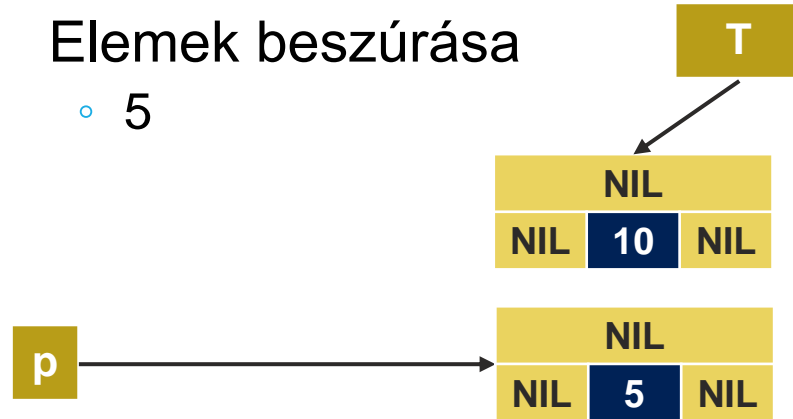
else  $\text{jobb}[y] \leftarrow p$



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

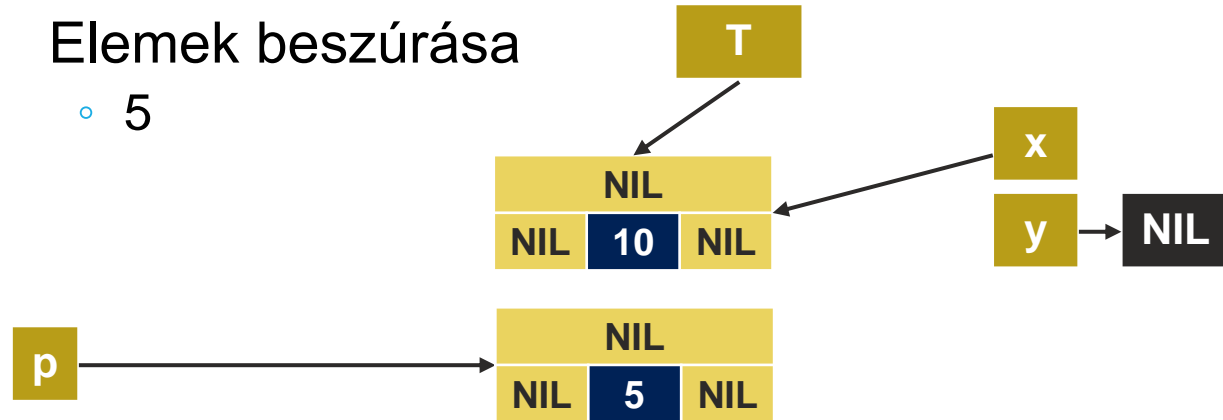
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

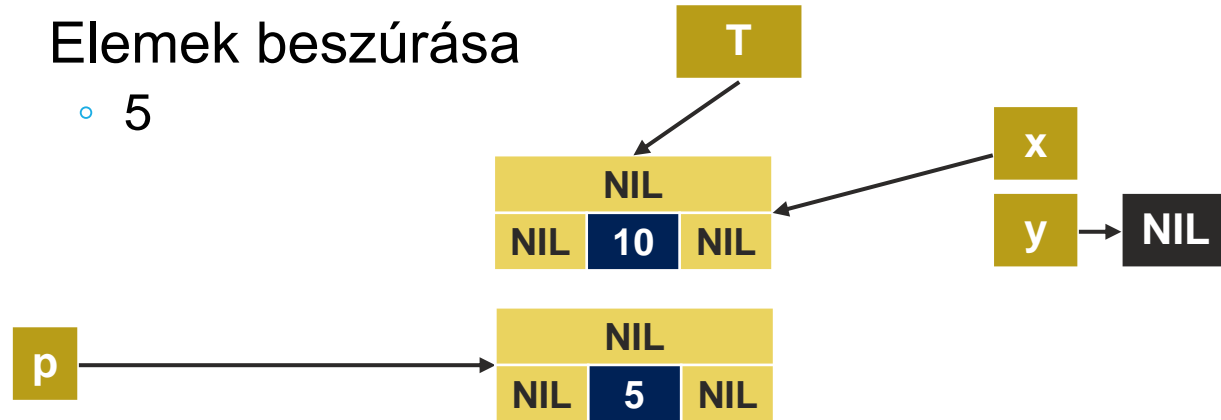
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

**do**  $y \leftarrow x$

**if**  $\text{kulcs}[p] < \text{kulcs}[x]$

**then**  $x \leftarrow \text{bal}[x]$

**else**  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

**if**  $y = \text{NIL}$

**then**  $\text{gyökér}[T] \leftarrow p$

**else if**  $\text{kulcs}[p] < \text{kulcs}[y]$

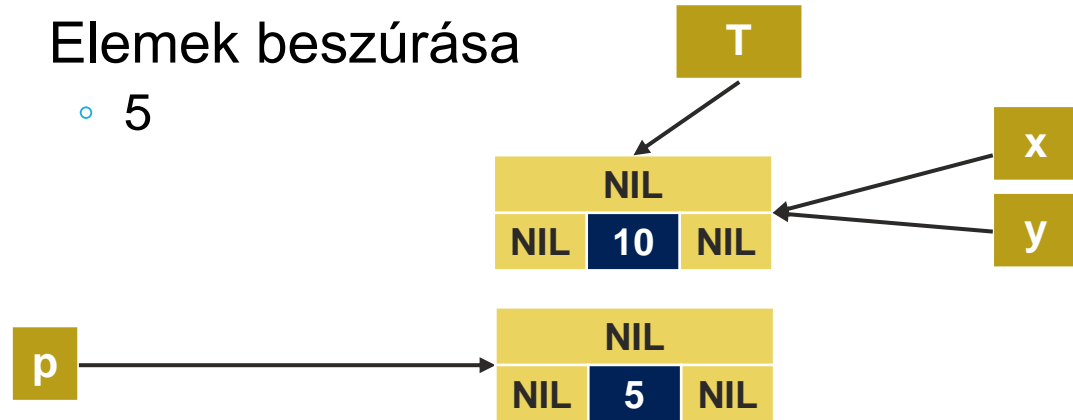
**then**  $\text{bal}[y] \leftarrow p$

**else**  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

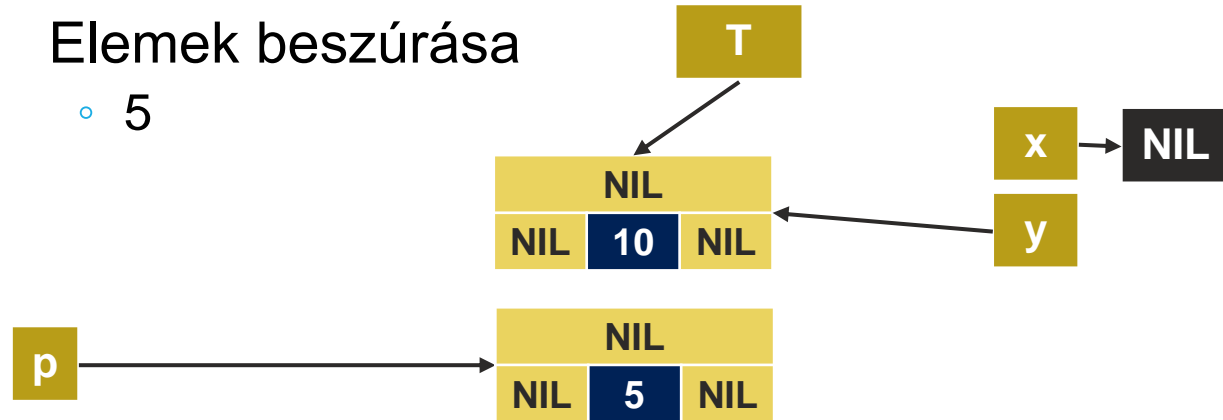
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

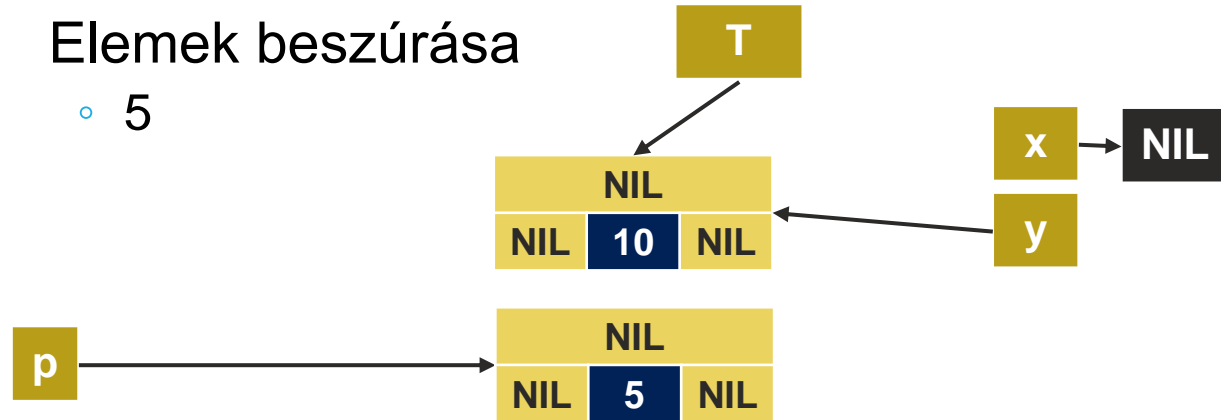
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

**if**  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else **if**  $\text{kulcs}[p] < \text{kulcs}[y]$

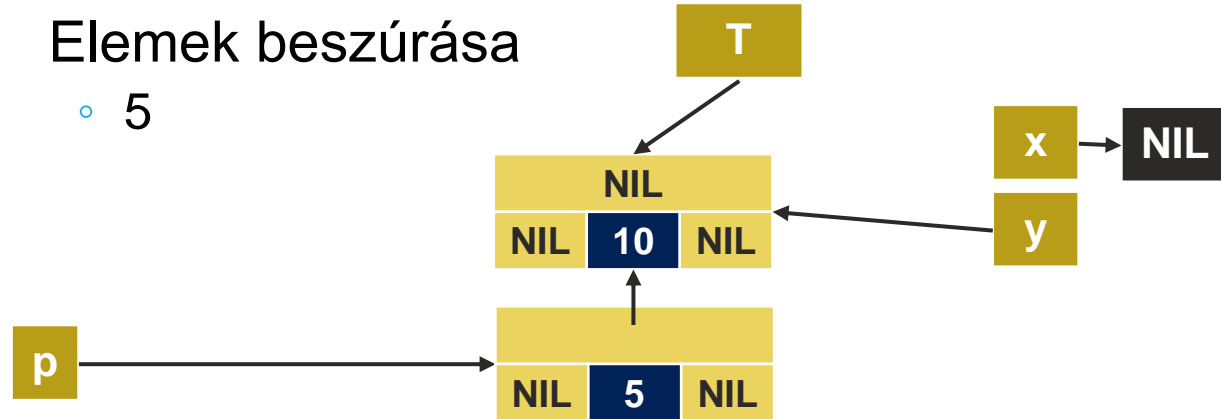
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

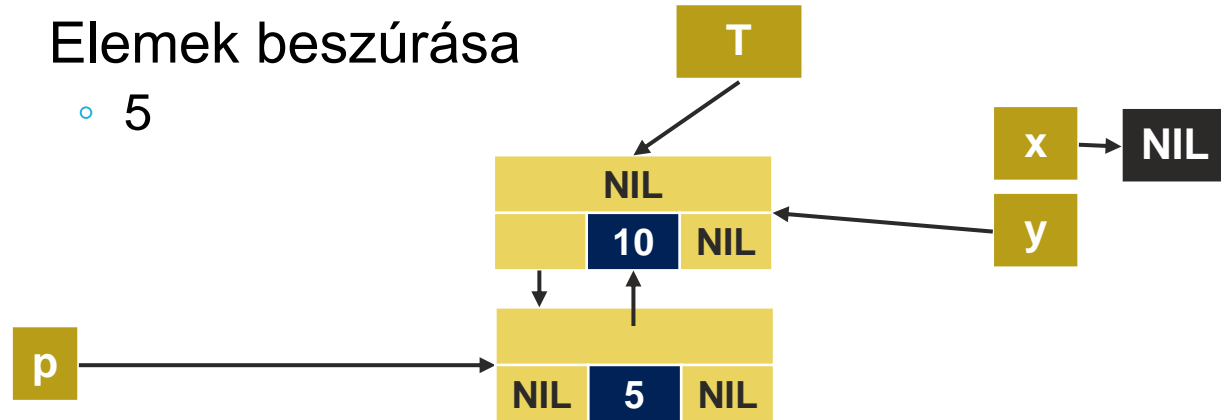
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 5



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

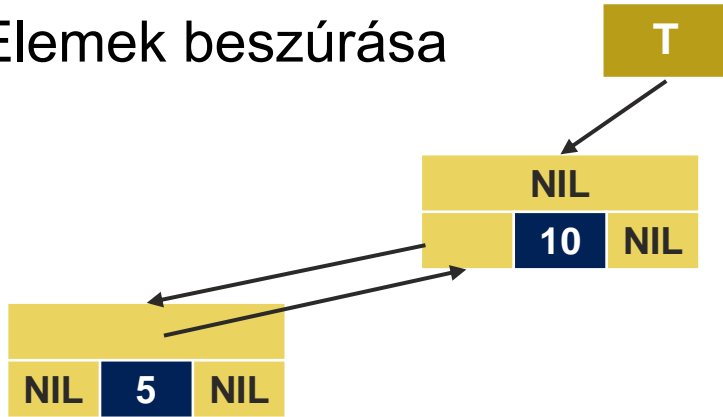
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

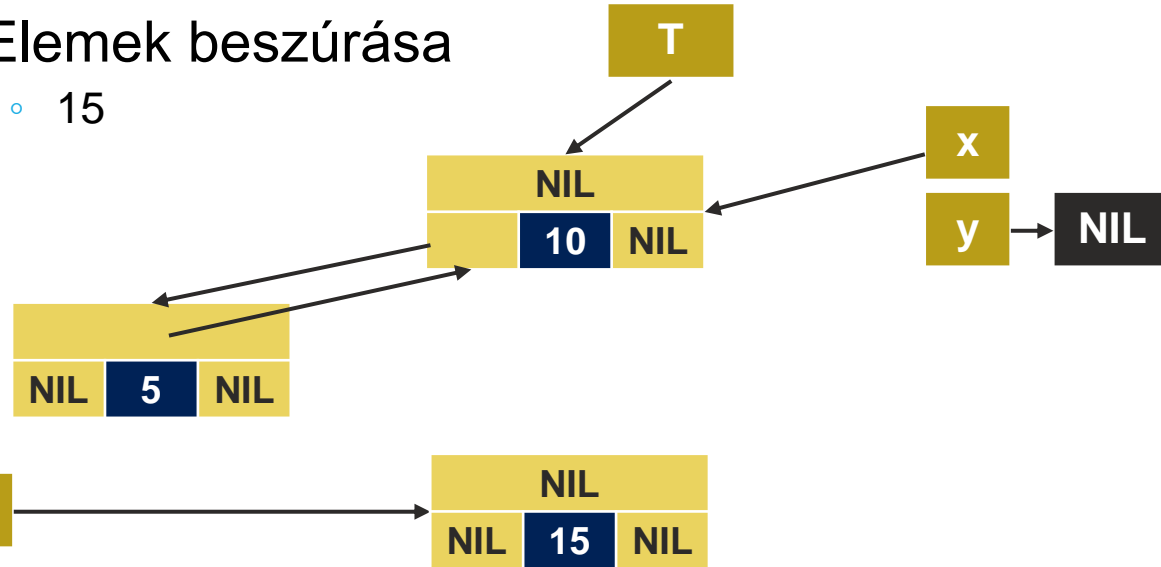
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

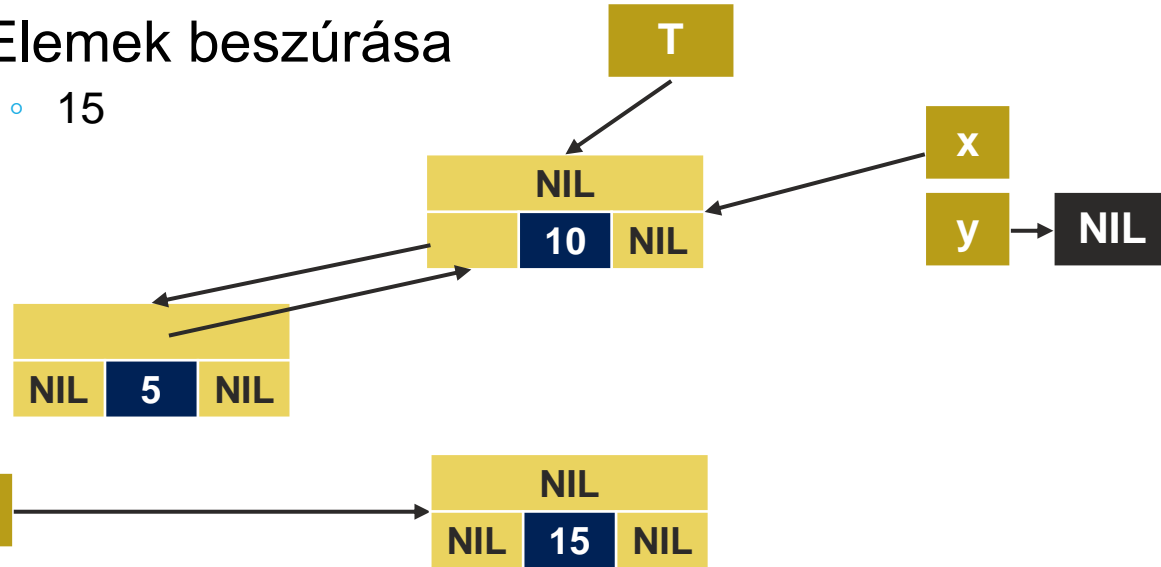
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

**do**  $y \leftarrow x$

**if**  $\text{kulcs}[p] < \text{kulcs}[x]$

**then**  $x \leftarrow \text{bal}[x]$

**else**  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

**if**  $y = \text{NIL}$

**then**  $\text{gyökér}[T] \leftarrow p$

**else if**  $\text{kulcs}[p] < \text{kulcs}[y]$

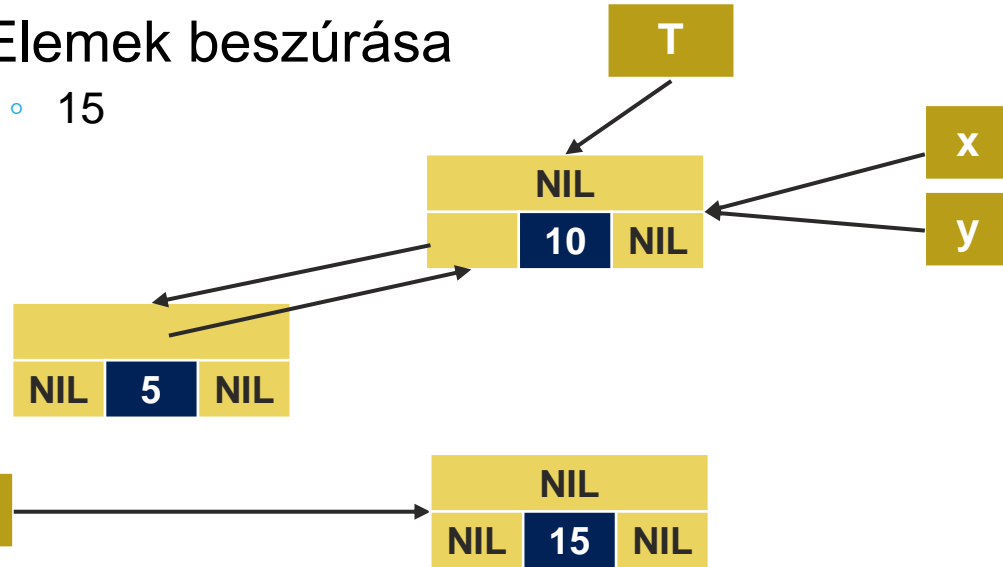
**then**  $\text{bal}[y] \leftarrow p$

**else**  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

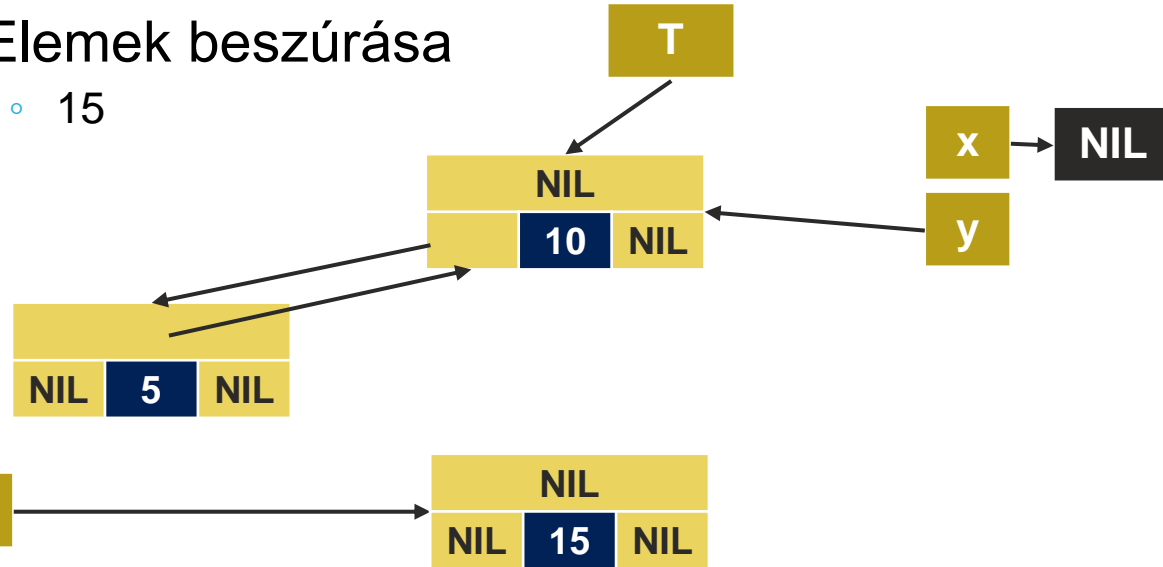
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

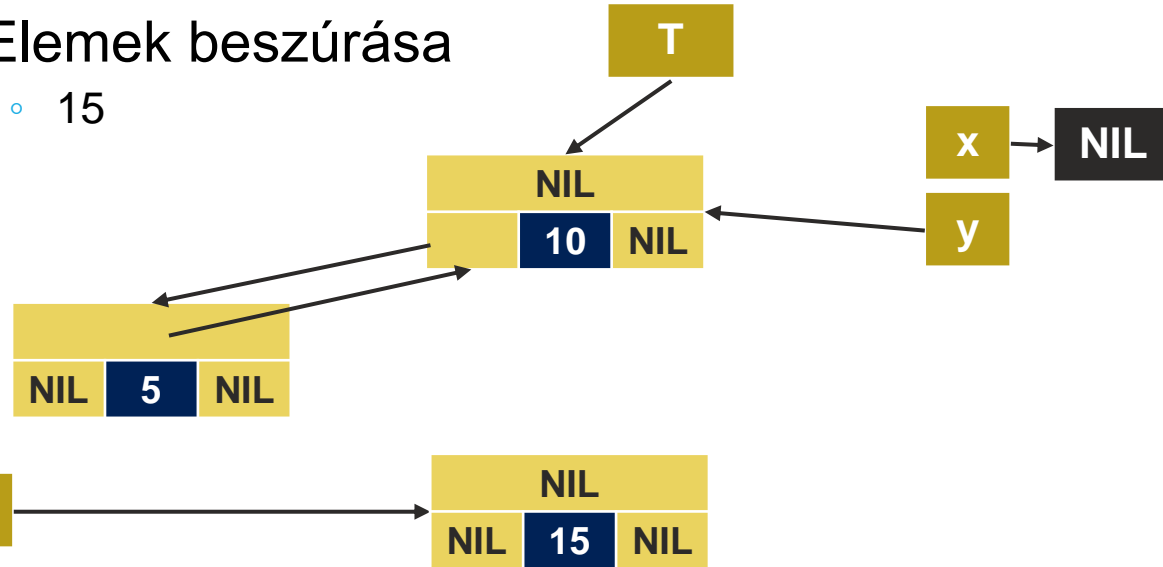
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

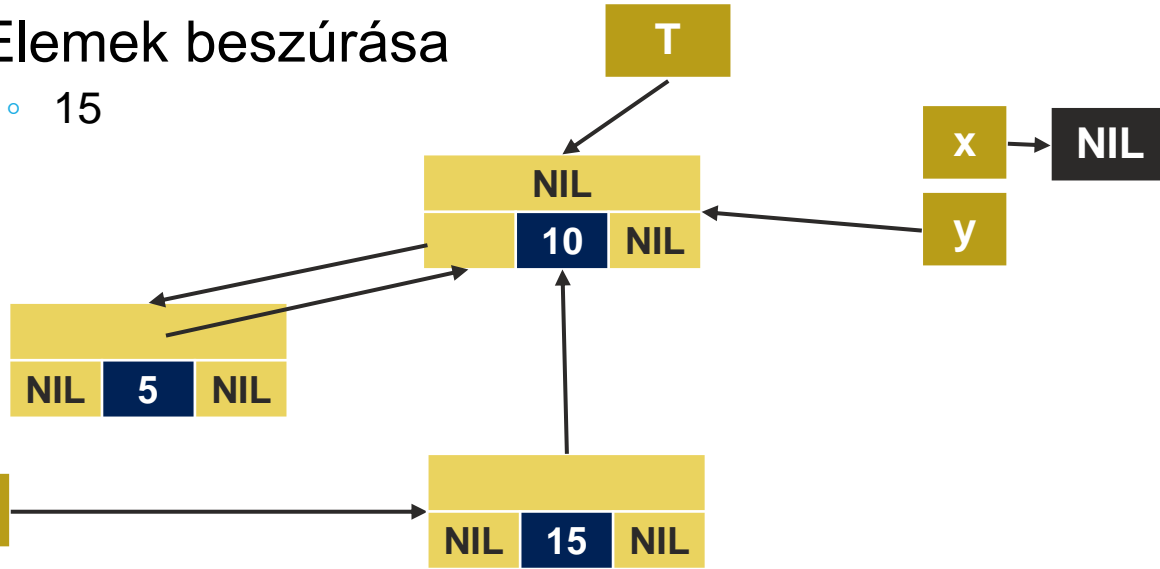
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

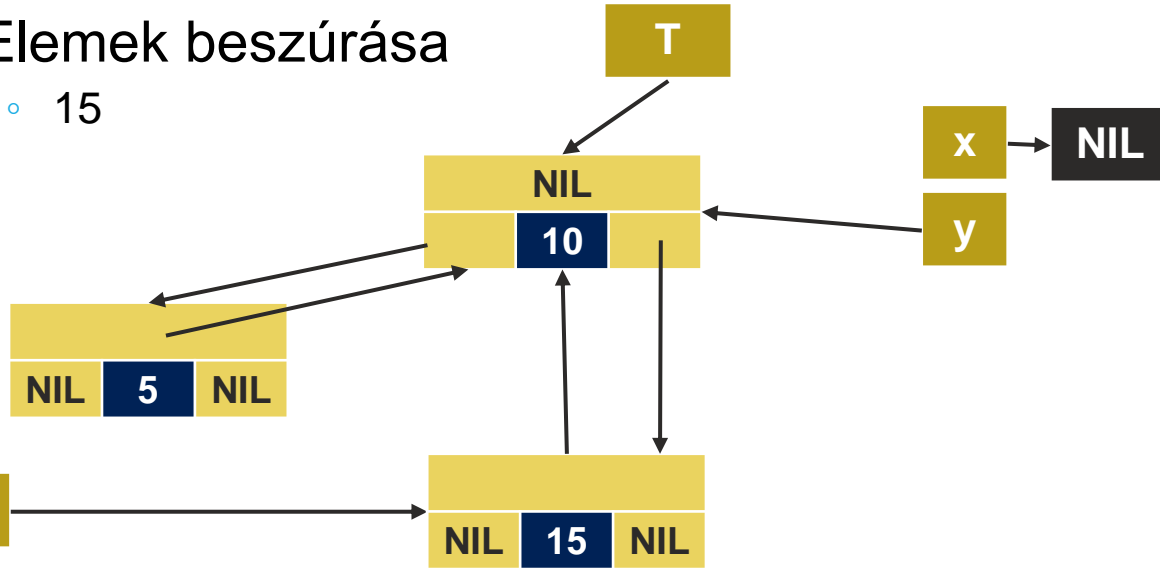
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 15



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

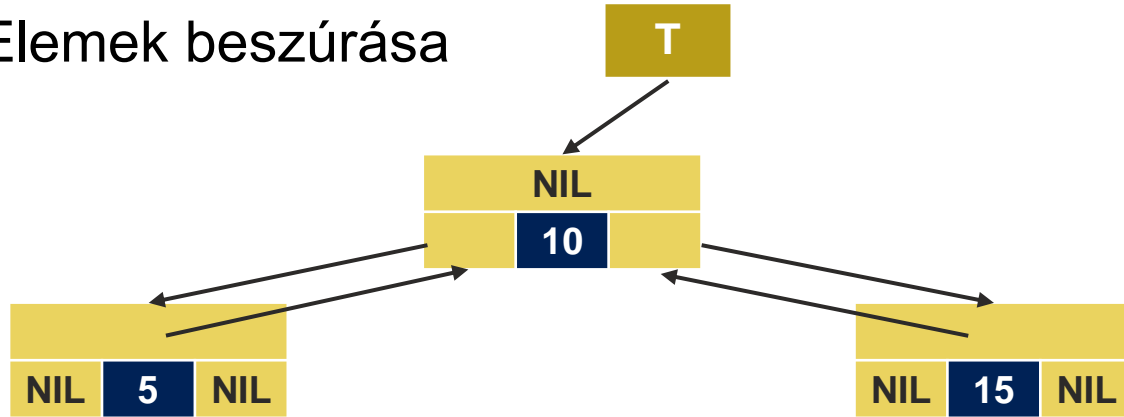
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

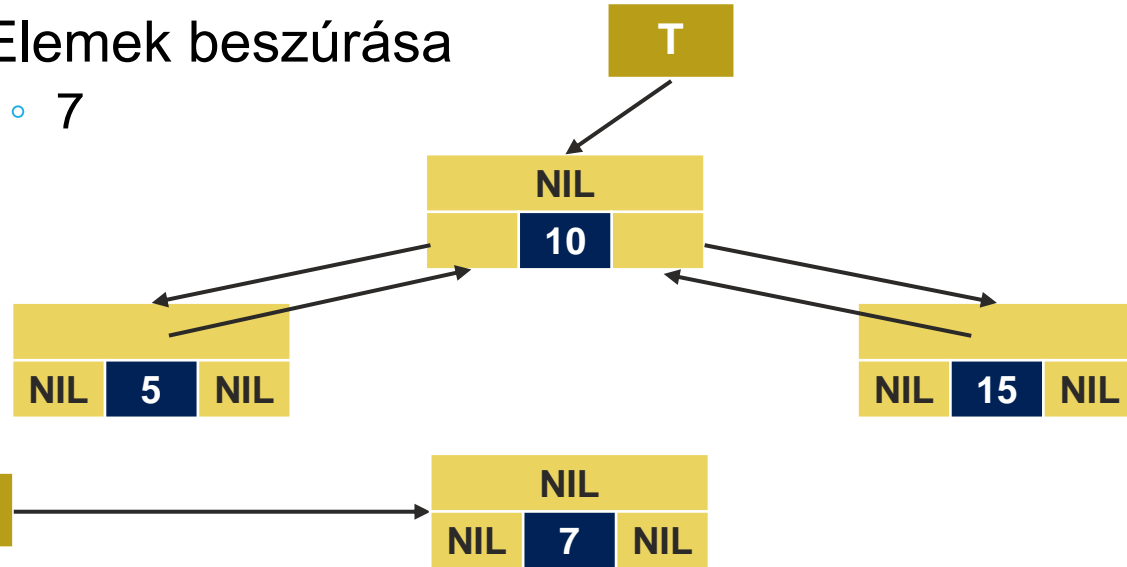
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

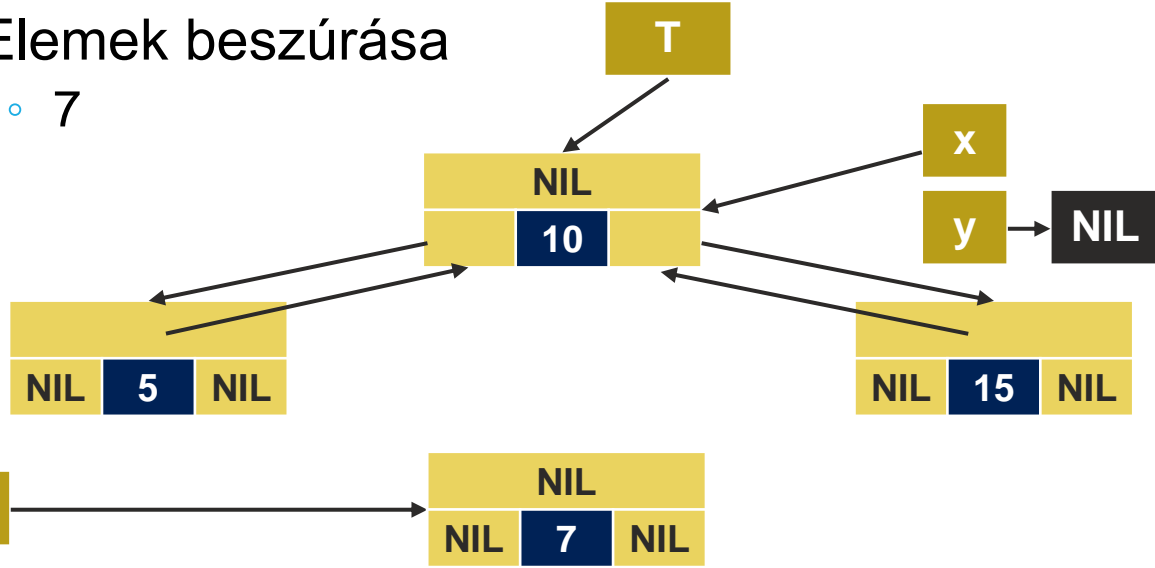
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr( $T, p$ )

# Elemek beszűrése

- 7



## Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

```
while x  $\neq$  NIL
```

do  $y \leftarrow x$

```
if kulcs[p] < kulcs[x]
```

```
then  $x \leftarrow \text{bal}[x]$ 
```

```
else x ← jobb[x]
```

$$\text{szülő}[p] \leftarrow y$$

```
if y = NIL
```

```
then gyökér[T] ← p
```

```
else if kulcs[p] < kulcs[y]
```

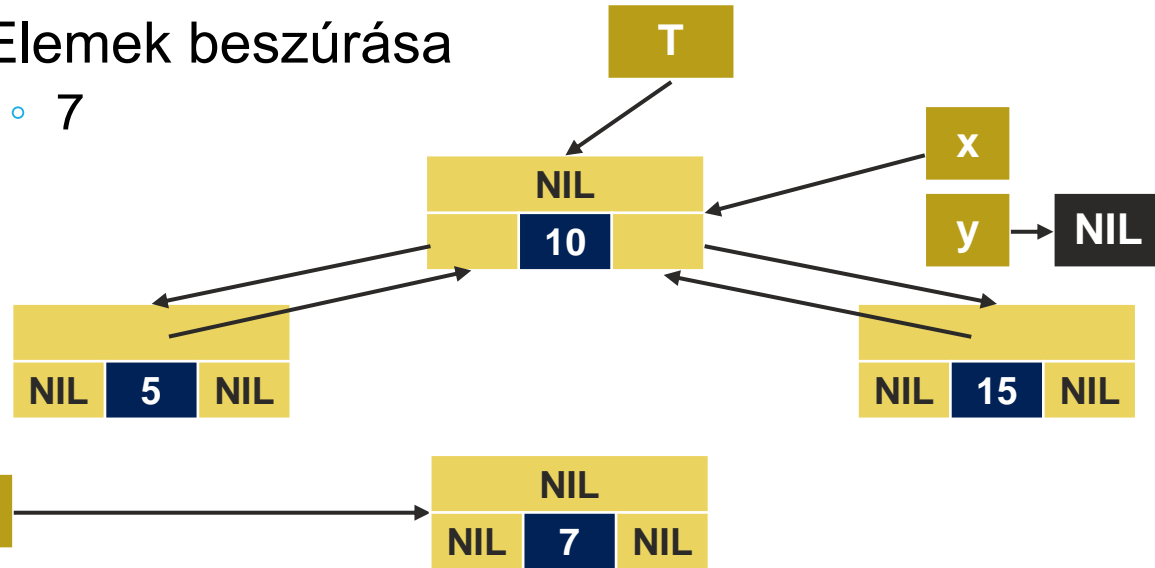
```
then bal[y]  $\leftarrow$  p
```

```
else jobb[y]  $\leftarrow$  p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

**do**  $y \leftarrow x$

**if**  $\text{kulcs}[p] < \text{kulcs}[x]$

**then**  $x \leftarrow \text{bal}[x]$

**else**  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

**if**  $y = \text{NIL}$

**then**  $\text{gyökér}[T] \leftarrow p$

**else if**  $\text{kulcs}[p] < \text{kulcs}[y]$

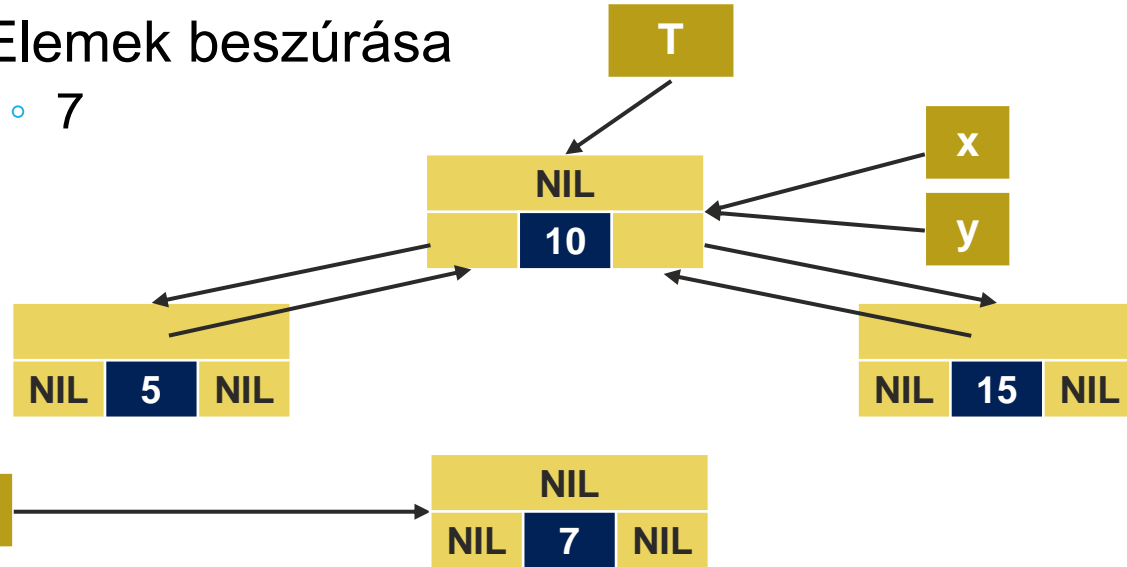
**then**  $\text{bal}[y] \leftarrow p$

**else**  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

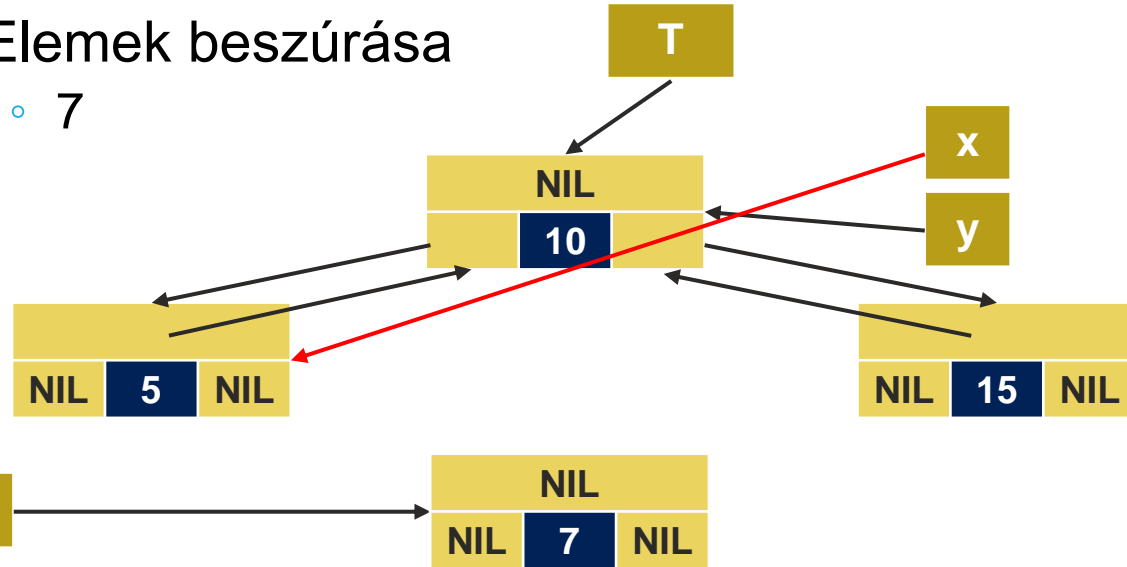
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

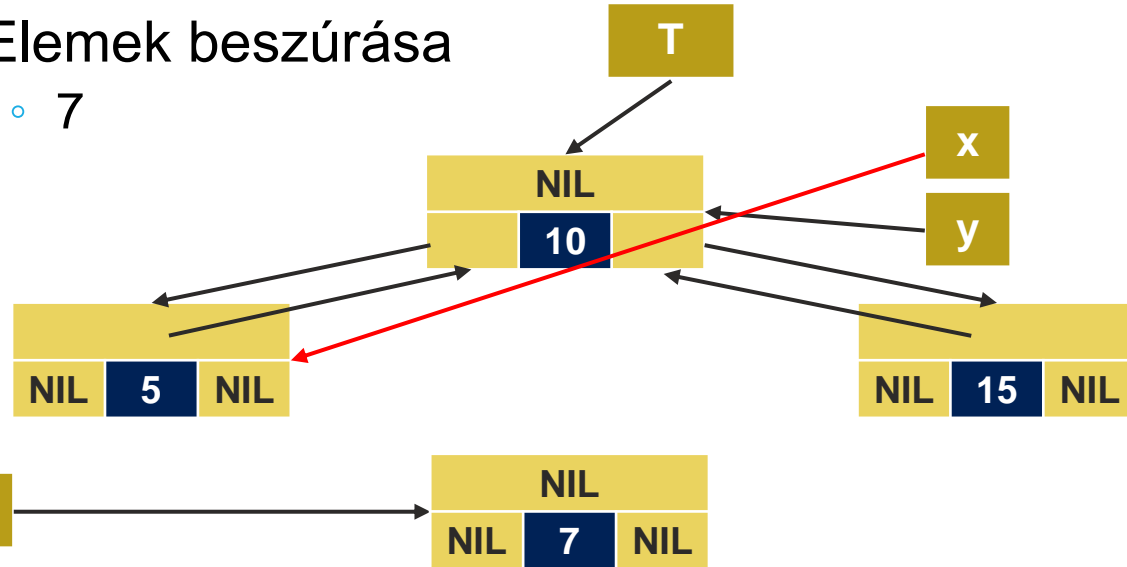
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

**do**  $y \leftarrow x$

**if**  $\text{kulcs}[p] < \text{kulcs}[x]$

**then**  $x \leftarrow \text{bal}[x]$

**else**  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

**if**  $y = \text{NIL}$

**then**  $\text{gyökér}[T] \leftarrow p$

**else if**  $\text{kulcs}[p] < \text{kulcs}[y]$

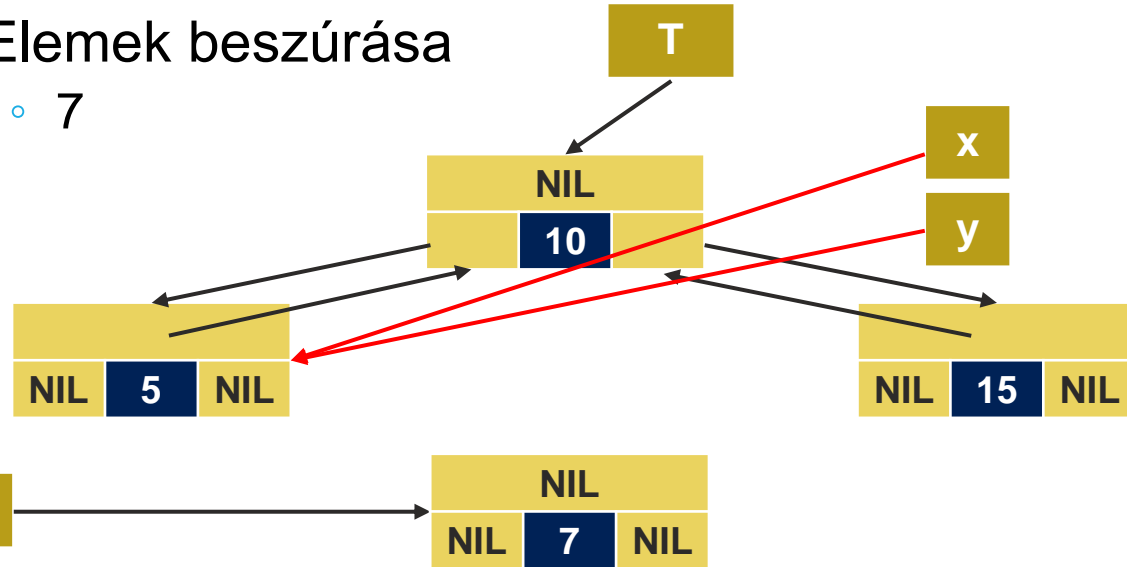
**then**  $\text{bal}[y] \leftarrow p$

**else**  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

then  $\text{bal}[y] \leftarrow p$

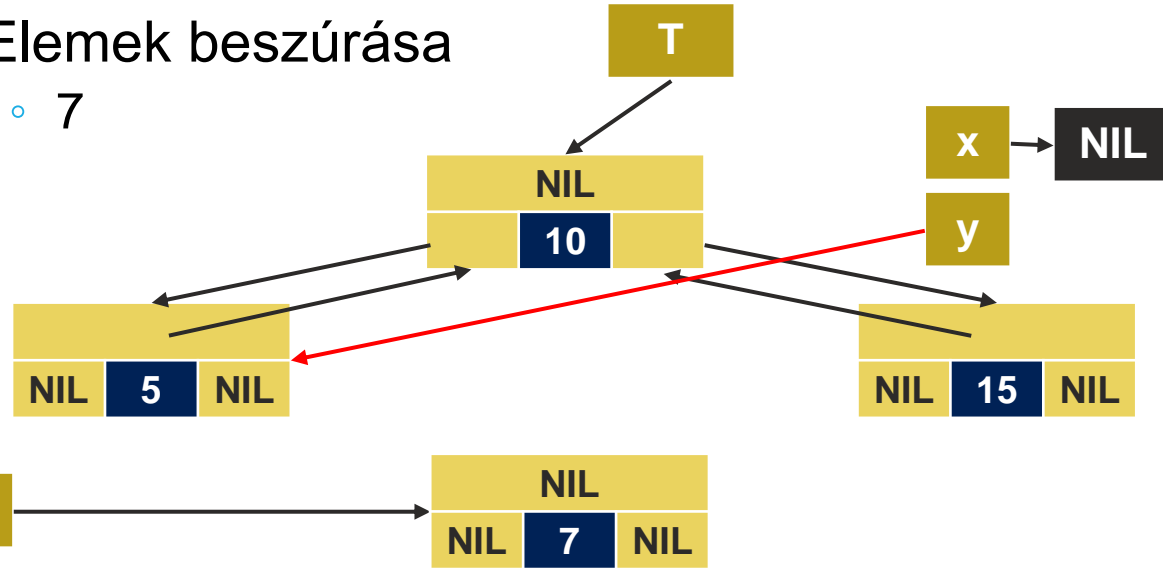
else  $\text{jobb}[y] \leftarrow p$



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

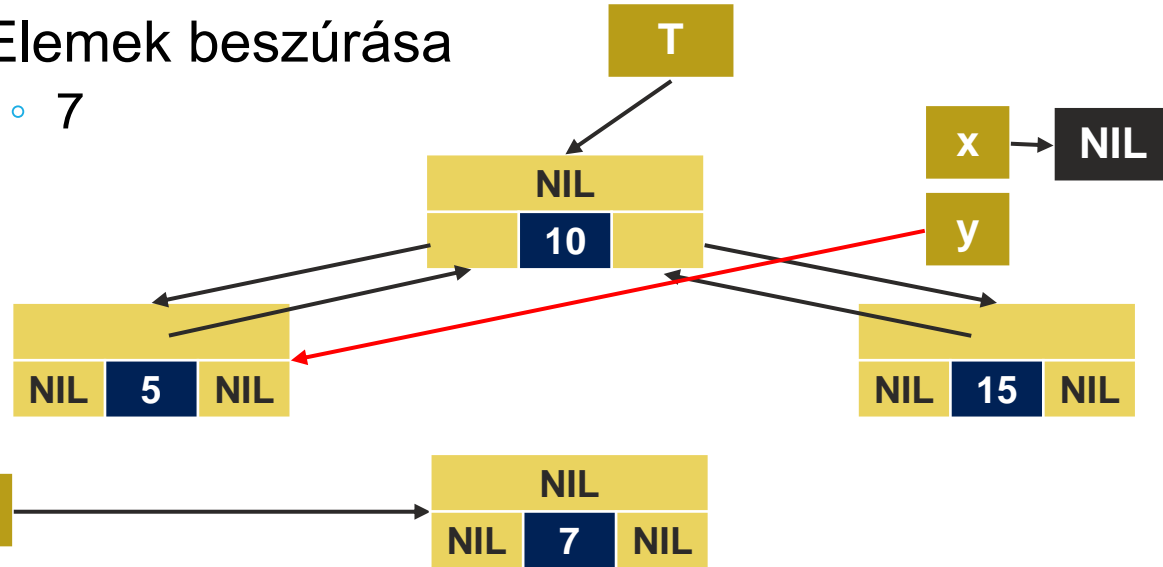
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

**while**  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

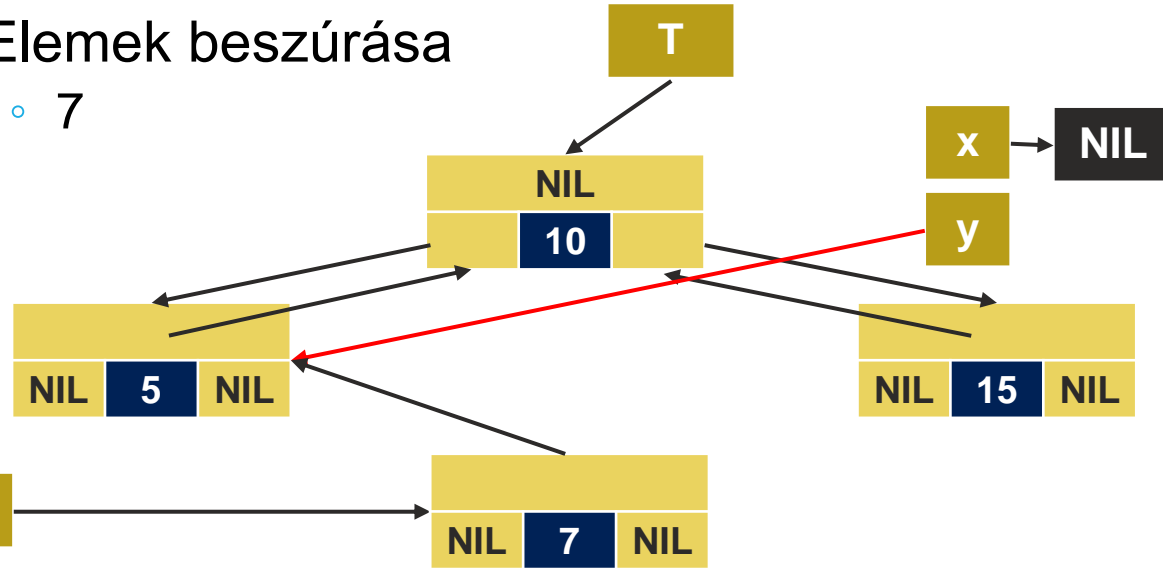
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

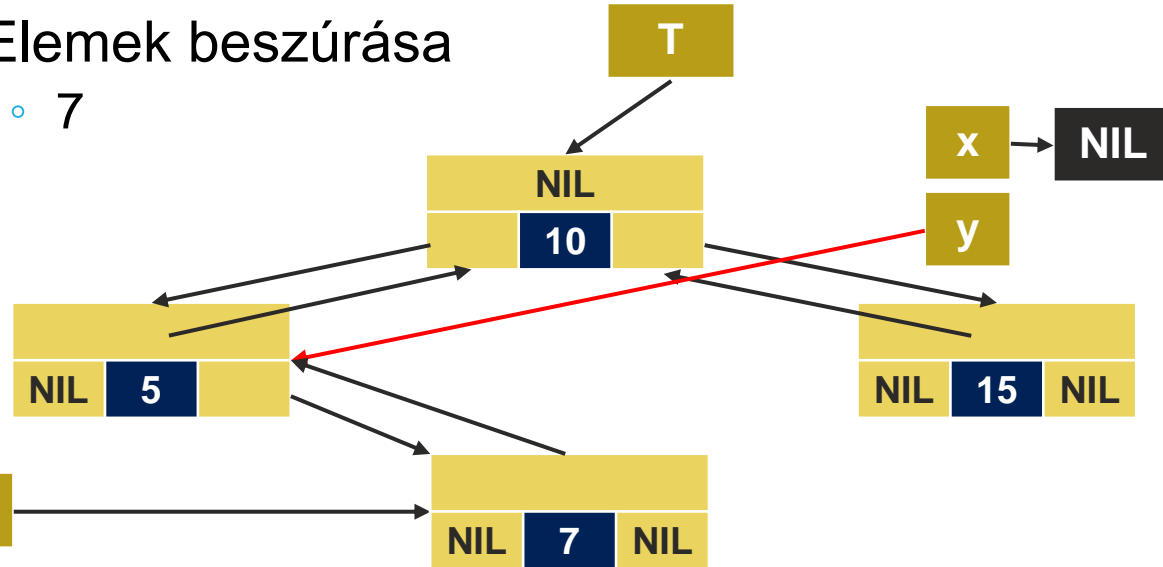
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 7



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

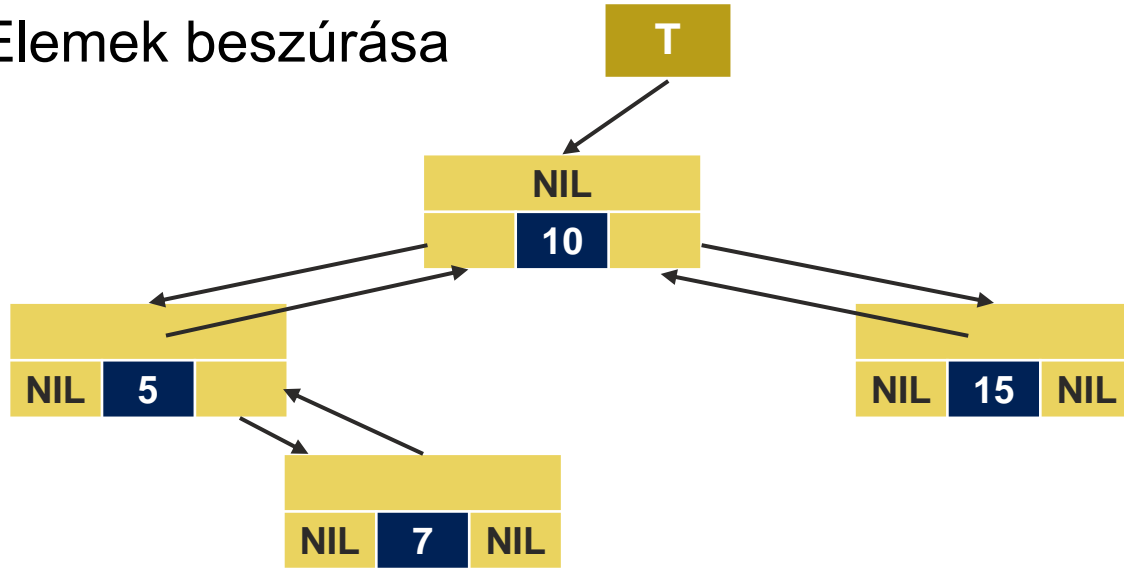
else if  $\text{kulcs}[p] < \text{kulcs}[y]$

then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

while  $x \neq \text{NIL}$

do  $y \leftarrow x$

if  $\text{kulcs}[p] < \text{kulcs}[x]$

then  $x \leftarrow \text{bal}[x]$

else  $x \leftarrow \text{jobb}[x]$

$\text{szülő}[p] \leftarrow y$

if  $y = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow p$

else if  $\text{kulcs}[p] < \text{kulcs}[y]$

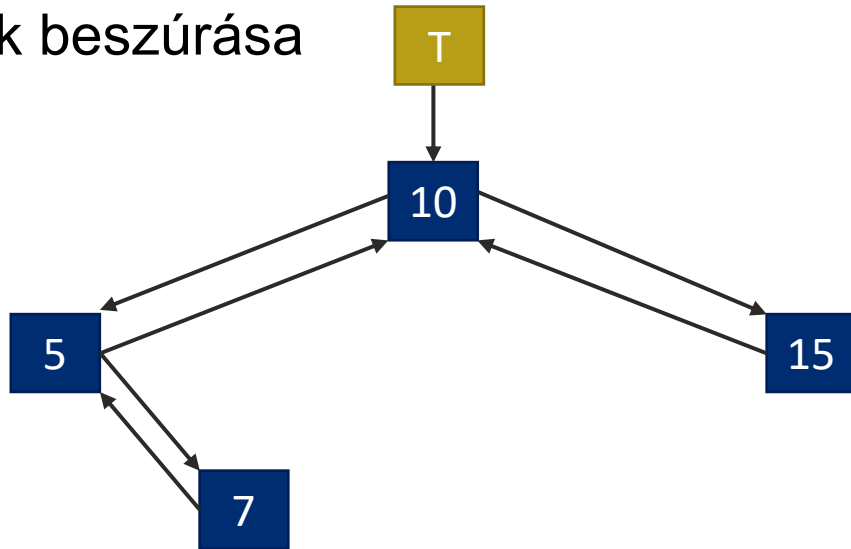
then  $\text{bal}[y] \leftarrow p$

else  $\text{jobb}[y] \leftarrow p$

A továbbiakban a fa csomópontjainak részletei az animációban nem lesznek feltüntetve.

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



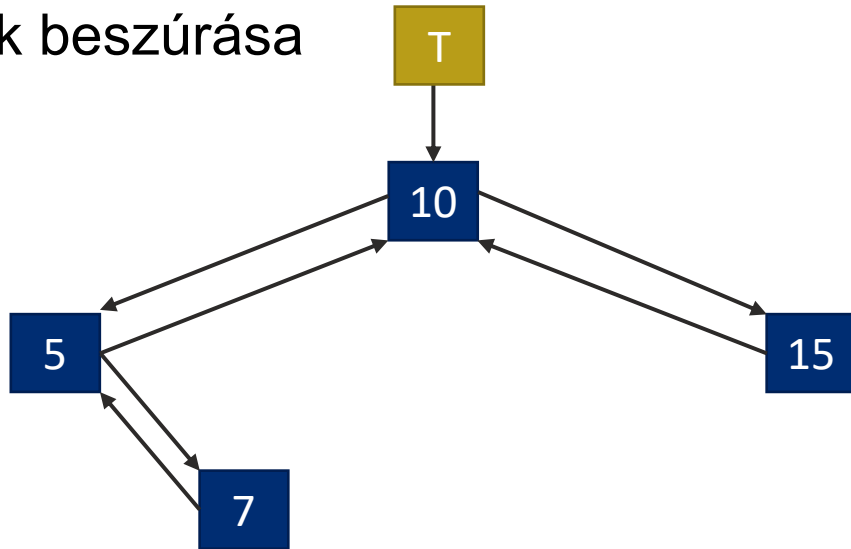
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
  do y ← x
    if kulcs[p] < kulcs[x]
      then x ← bal[x]
    else x ← jobb[x]
szülő[p] ← y
if y = NIL
  then gyökér[T] ← p
  else if kulcs[p] < kulcs[y]
    then bal[y] ← p
    else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



p 6

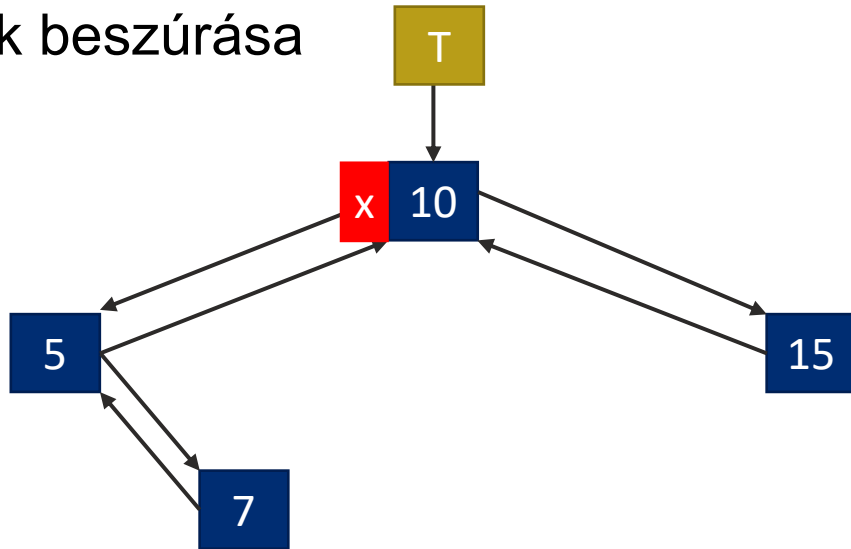
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



Fába-beszúr (T,p)

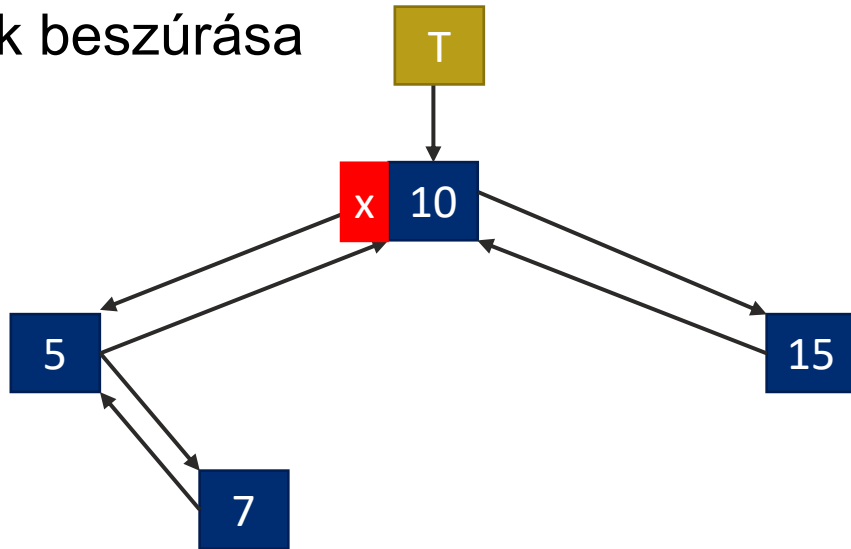
```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



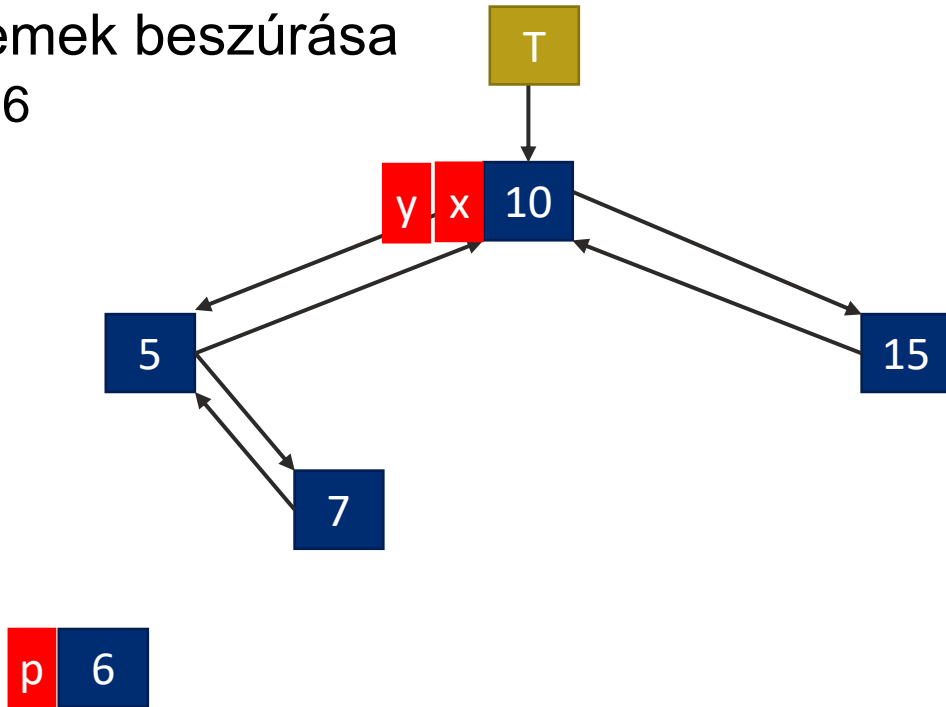
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



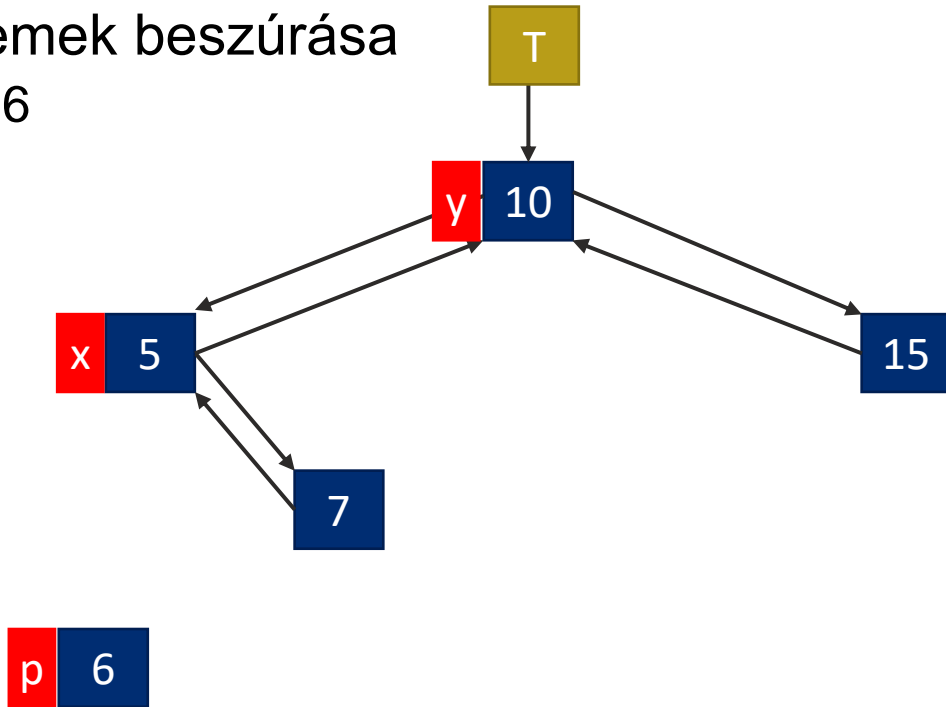
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



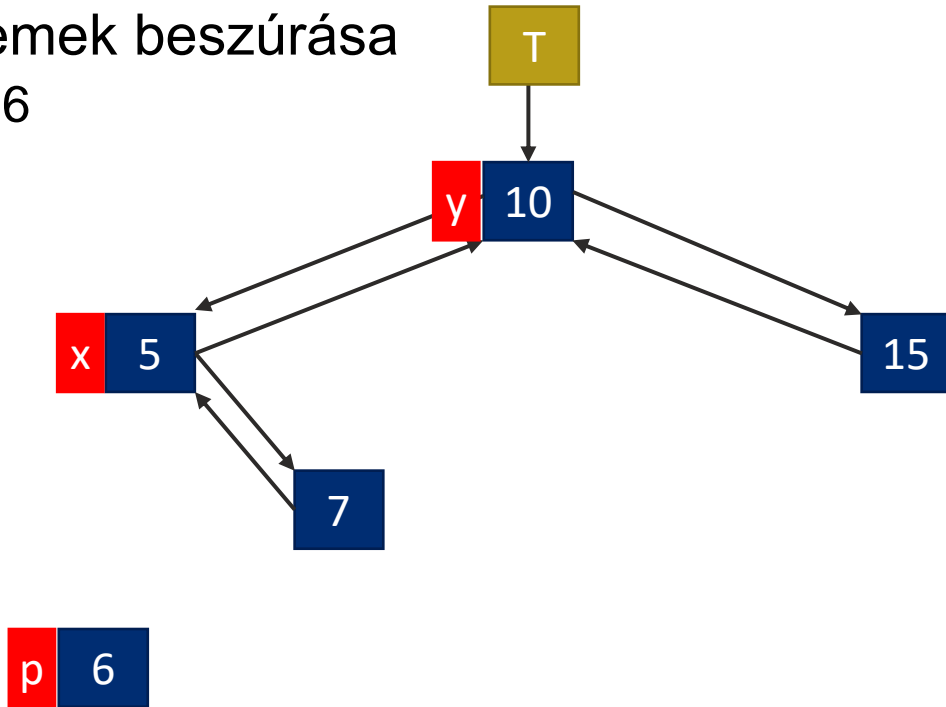
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



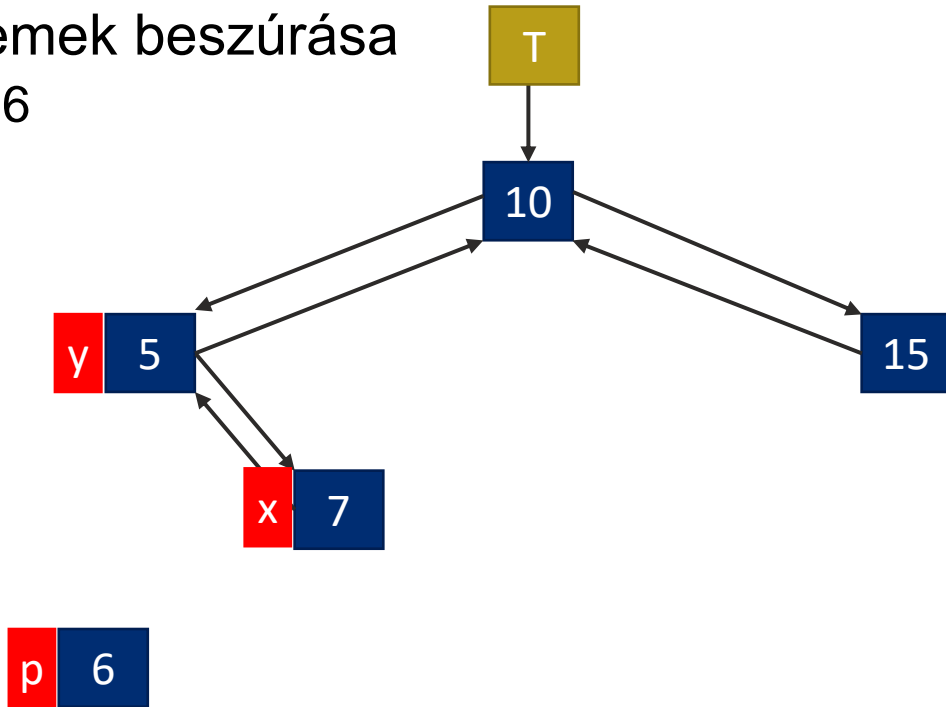
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



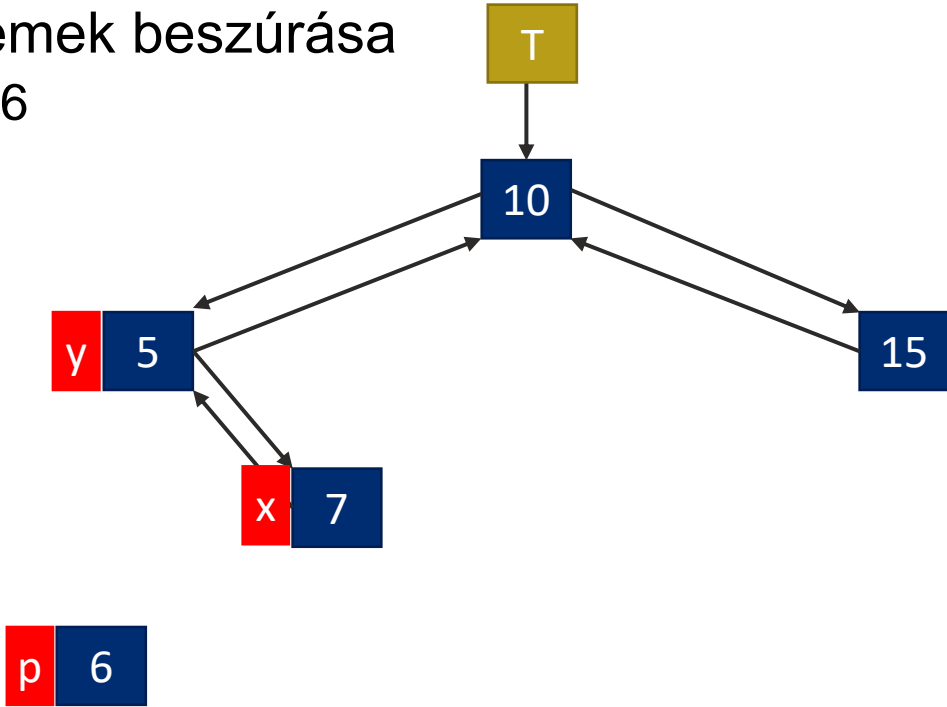
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr( $T, p$ )

# Elemek beszűrése

- 6



## Fába-beszúr (T,p)

$y \leftarrow \text{NIL}; x \leftarrow \text{gyökér}[T]$

```
while x ≠ NIL
```

do  $y \leftarrow x$

```
if kulcs[p] < kulcs[x]
```

```
then  $x \leftarrow \text{bal}[x]$ 
```

```
else x ← jobb[x]
```

$$\text{szülő}[p] \leftarrow y$$

```
if y = NIL
```

then  $\text{gyökér}[T] \leftarrow p$

```
else if kulcs[p] < kulcs[y]
```

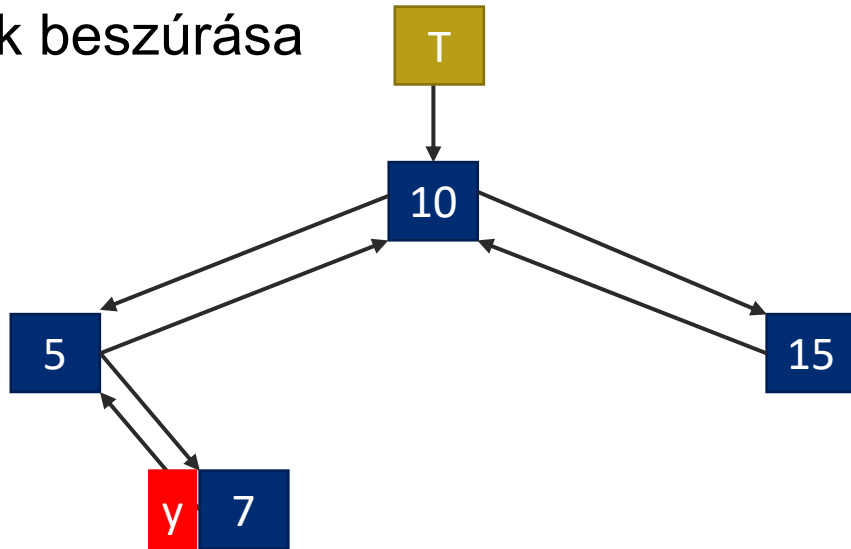
```
then bal[y]  $\leftarrow$  p
```

```
else jobb[y]  $\leftarrow$  p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



p 6

x

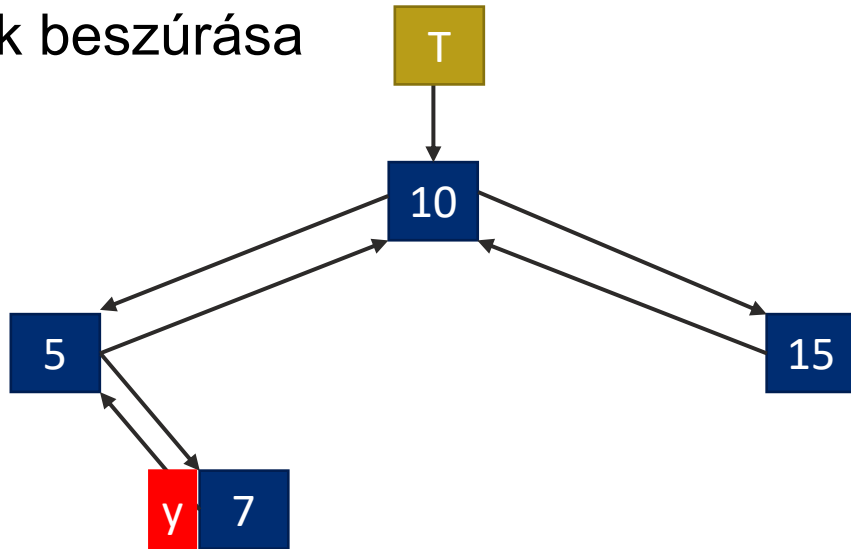
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



p 6

x

Fába-beszúr (T,p)

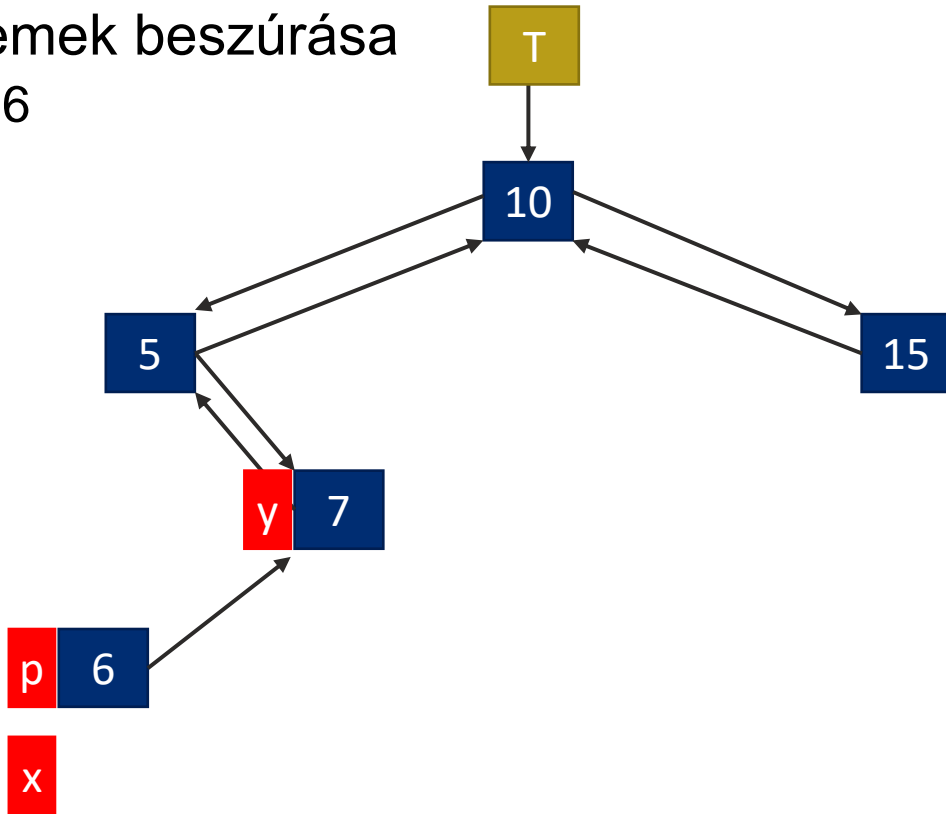
```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6



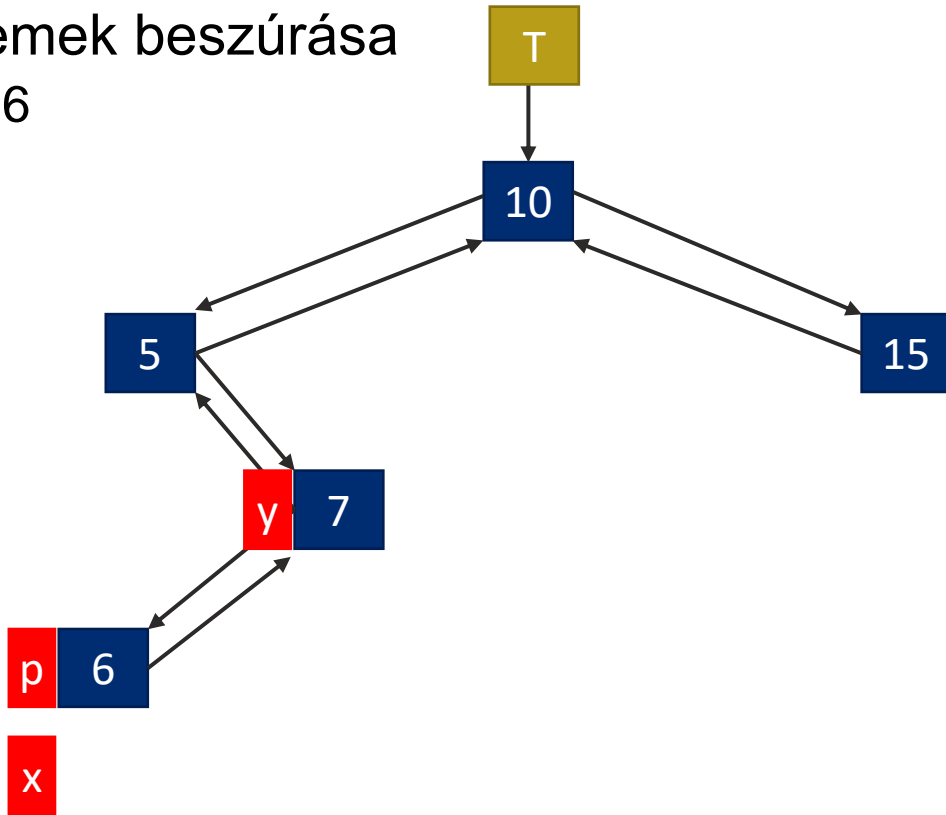
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása

- 6

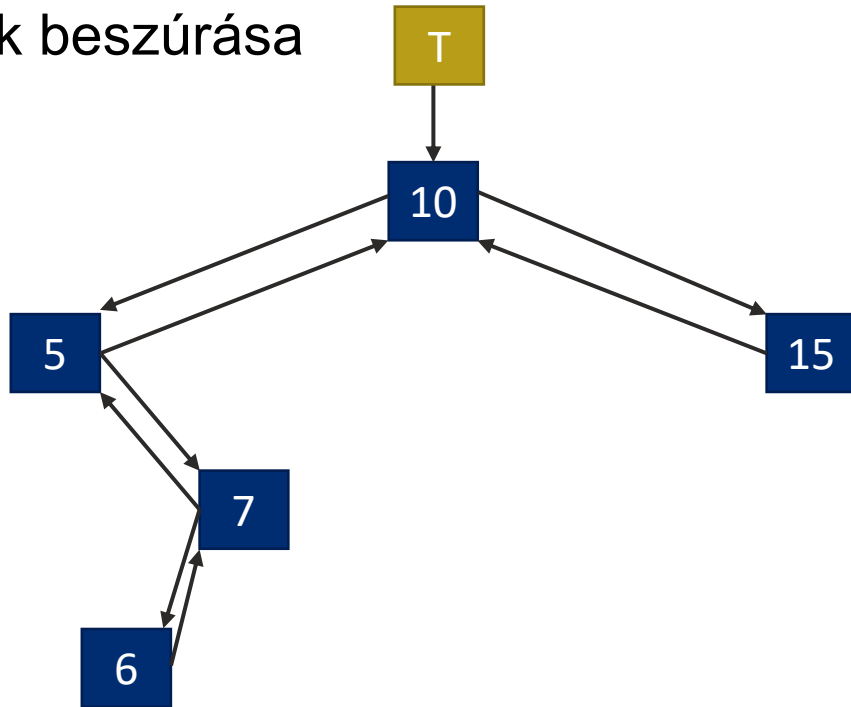


Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



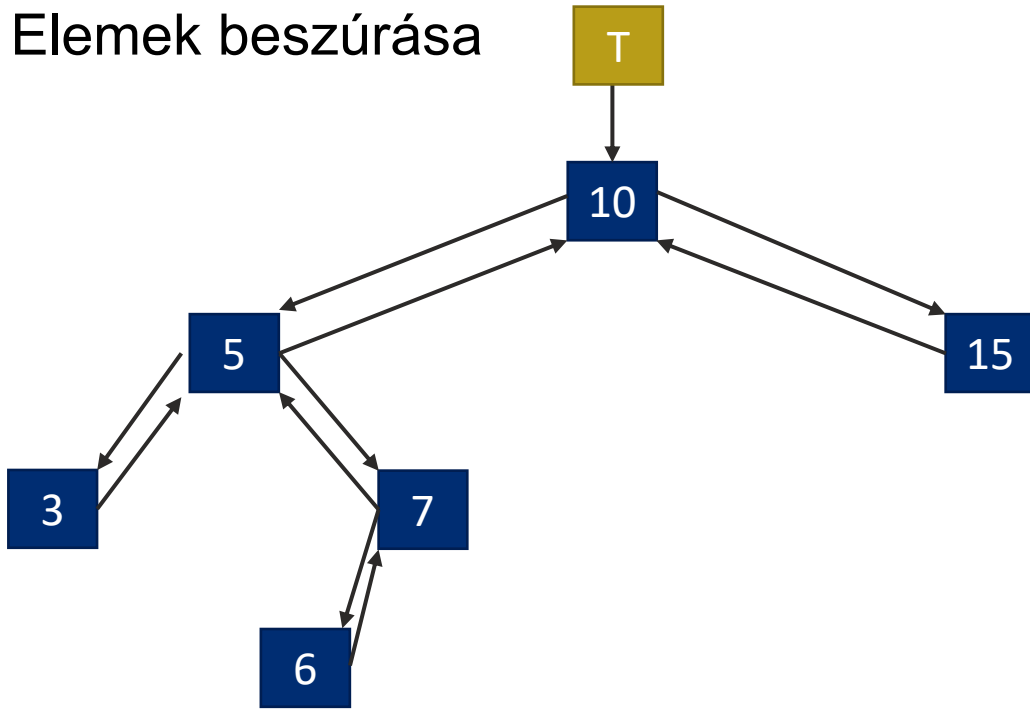
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 13

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



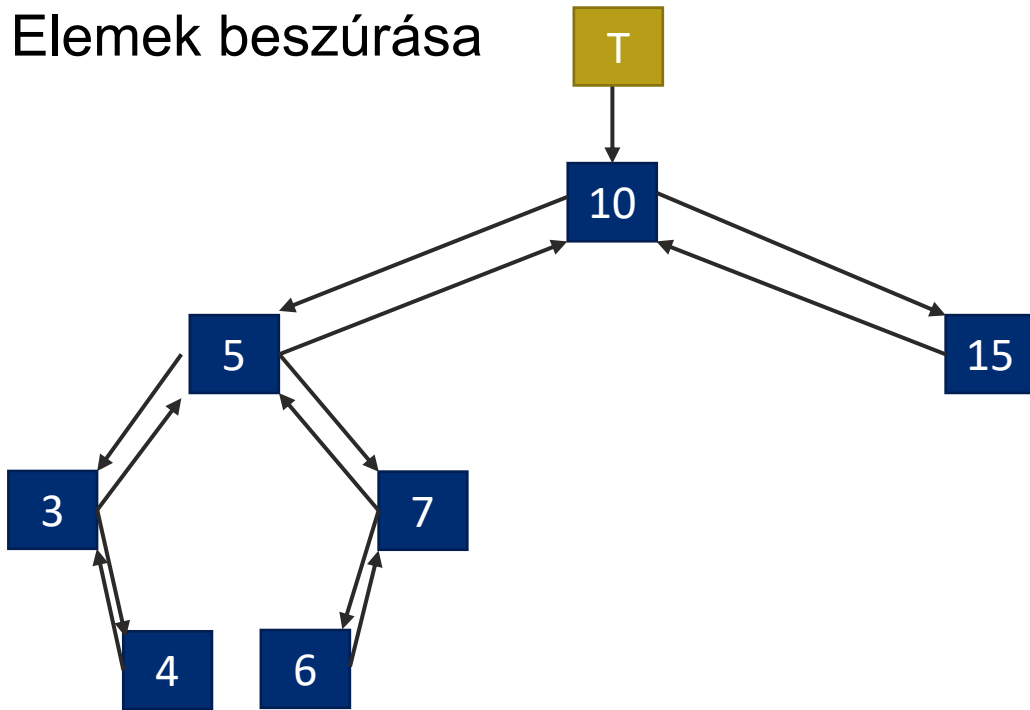
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
do y ← x
  if kulcs[p] < kulcs[x]
  then x ← bal[x]
  else x ← jobb[x]
szülő[p] ← y
if y = NIL
then gyökér[T] ← p
else if kulcs[p] < kulcs[y]
then bal[y] ← p
else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 21

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



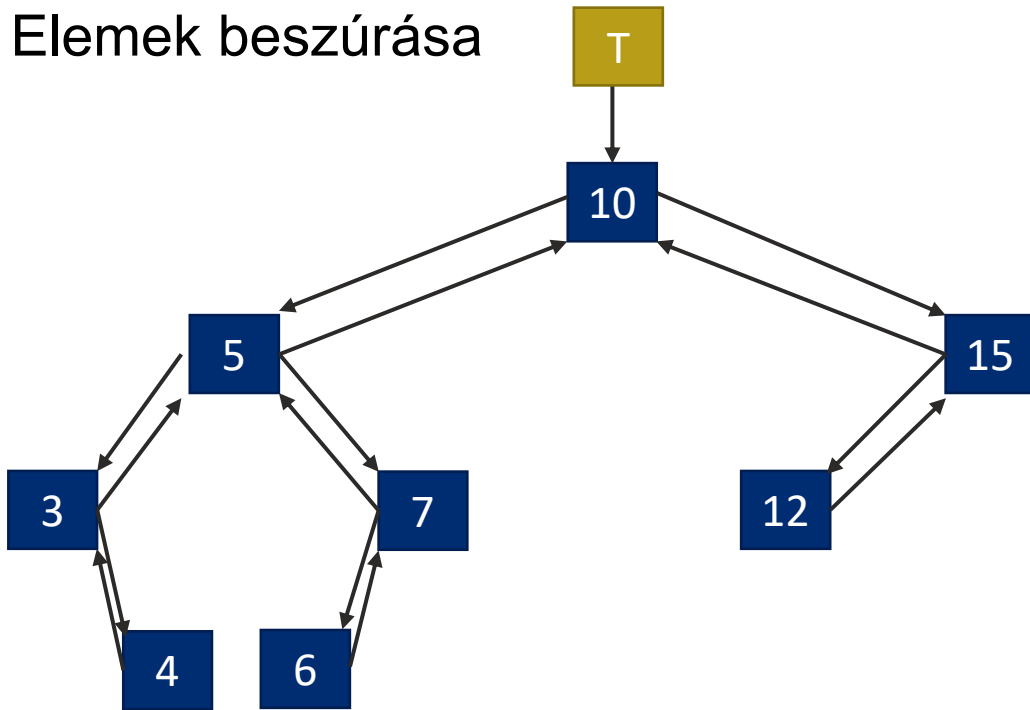
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
  do y ← x
    if kulcs[p] < kulcs[x]
      then x ← bal[x]
    else x ← jobb[x]
szülő[p] ← y
if y = NIL
  then gyökér[T] ← p
  else if kulcs[p] < kulcs[y]
    then bal[y] ← p
    else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 21

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



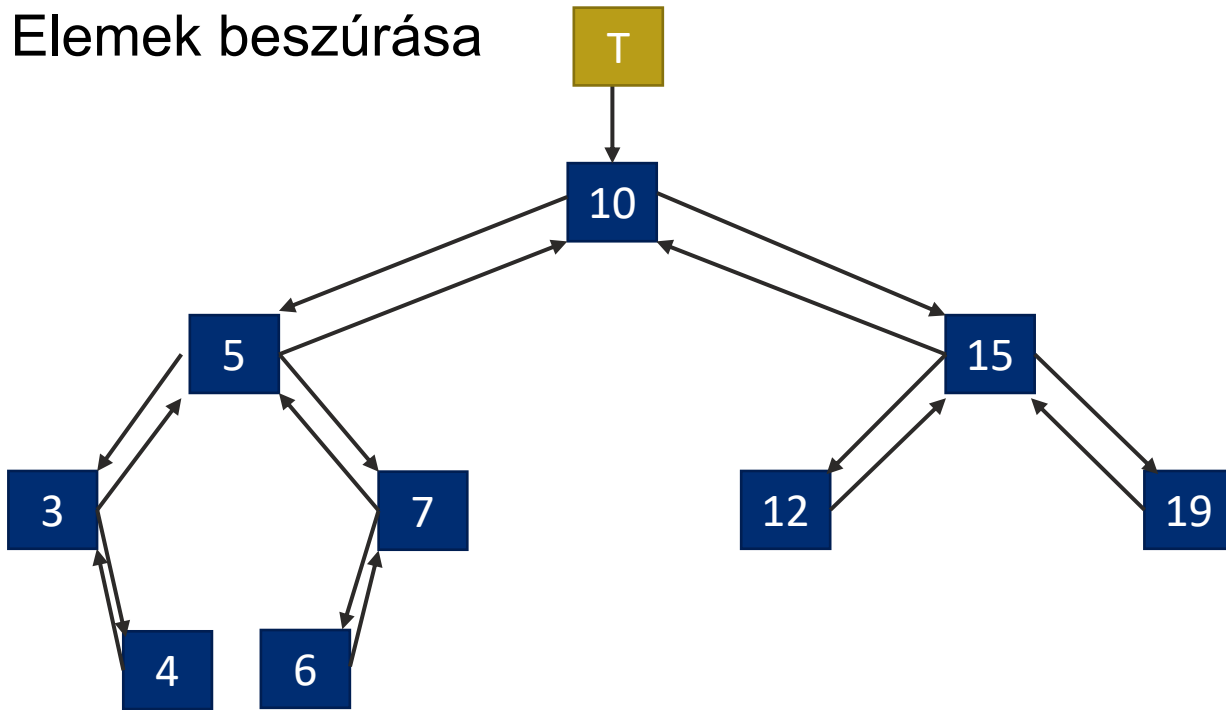
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
  do y ← x
    if kulcs[p] < kulcs[x]
      then x ← bal[x]
    else x ← jobb[x]
szülő[p] ← y
if y = NIL
  then gyökér[T] ← p
  else if kulcs[p] < kulcs[y]
    then bal[y] ← p
    else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 21

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



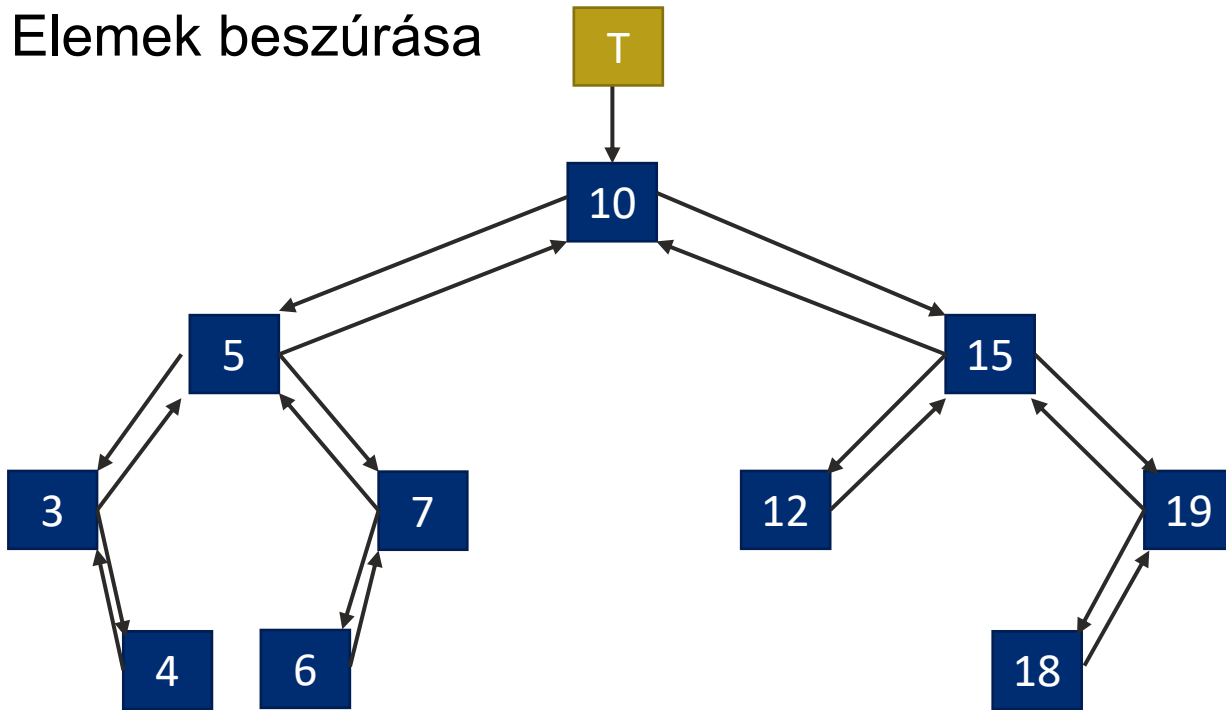
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
  do y ← x
    if kulcs[p] < kulcs[x]
      then x ← bal[x]
    else x ← jobb[x]
szülő[p] ← y
if y = NIL
  then gyökér[T] ← p
  else if kulcs[p] < kulcs[y]
    then bal[y] ← p
    else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 21

# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



Fába-beszúr (T,p)

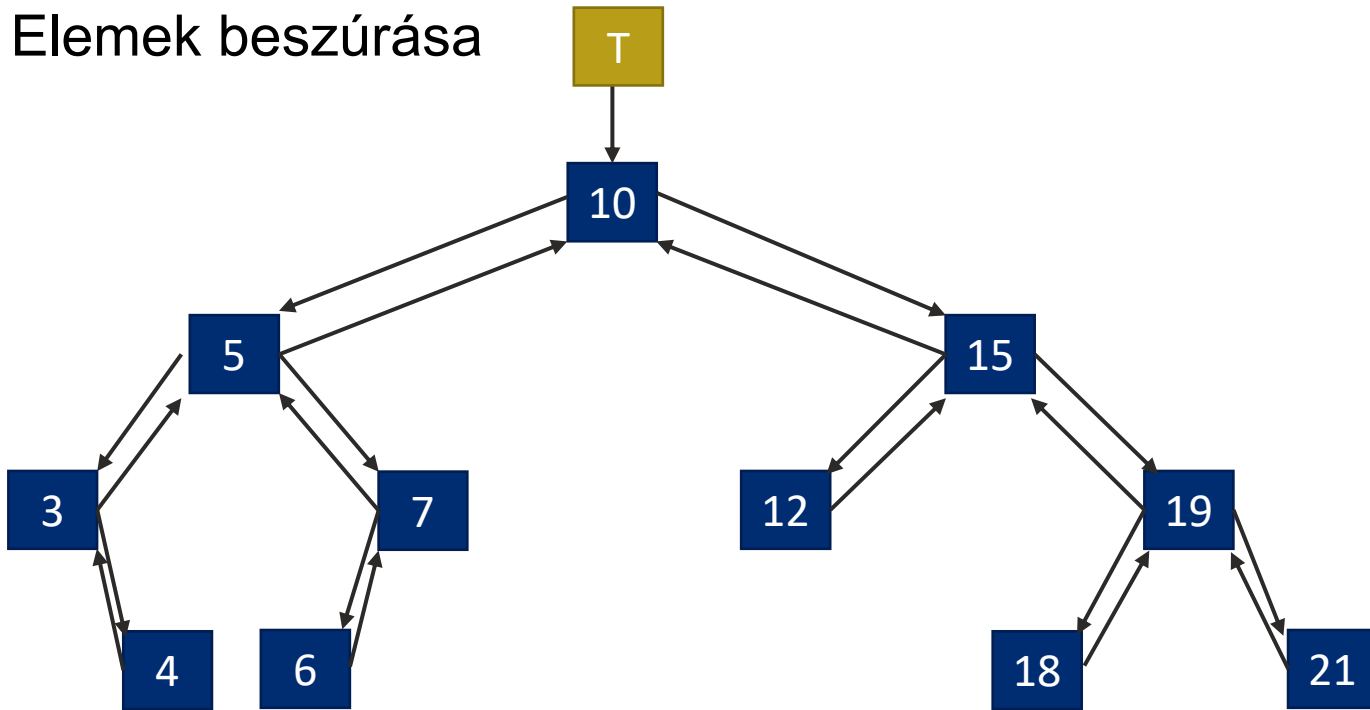
```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
  do y ← x
    if kulcs[p] < kulcs[x]
      then x ← bal[x]
    else x ← jobb[x]
szülő[p] ← y
if y = NIL
  then gyökér[T] ← p
  else if kulcs[p] < kulcs[y]
    then bal[y] ← p
    else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 21



# Bináris keresési fa – Fába-beszúr(T,p)

Elemek beszúrása



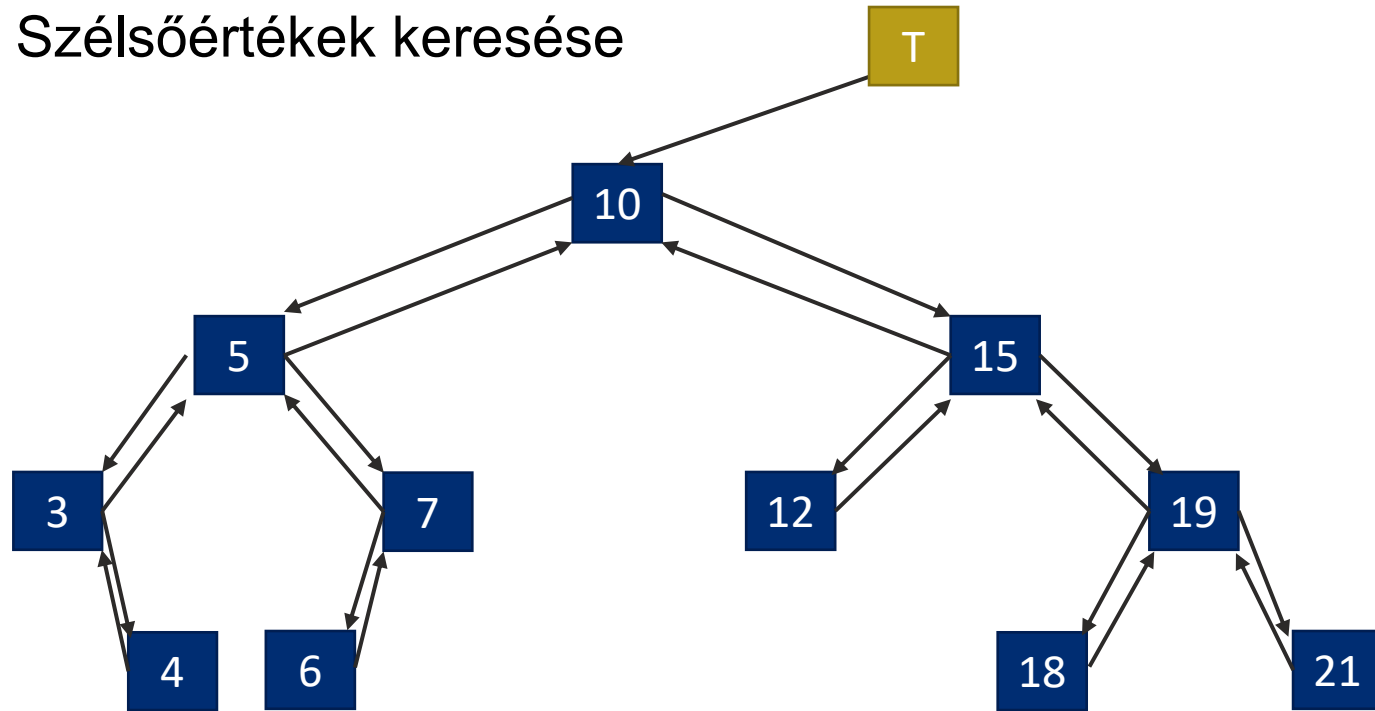
Fába-beszúr (T,p)

```
y ← NIL; x ← gyökér[T]
while x ≠ NIL
  do y ← x
    if kulcs[p] < kulcs[x]
      then x ← bal[x]
    else x ← jobb[x]
szülő[p] ← y
if y = NIL
  then gyökér[T] ← p
  else if kulcs[p] < kulcs[y]
    then bal[y] ← p
    else jobb[y] ← p
```

Ugyanezzel az algoritmussal még beszúrjuk: 3, 4, 12, 19, 18, 21

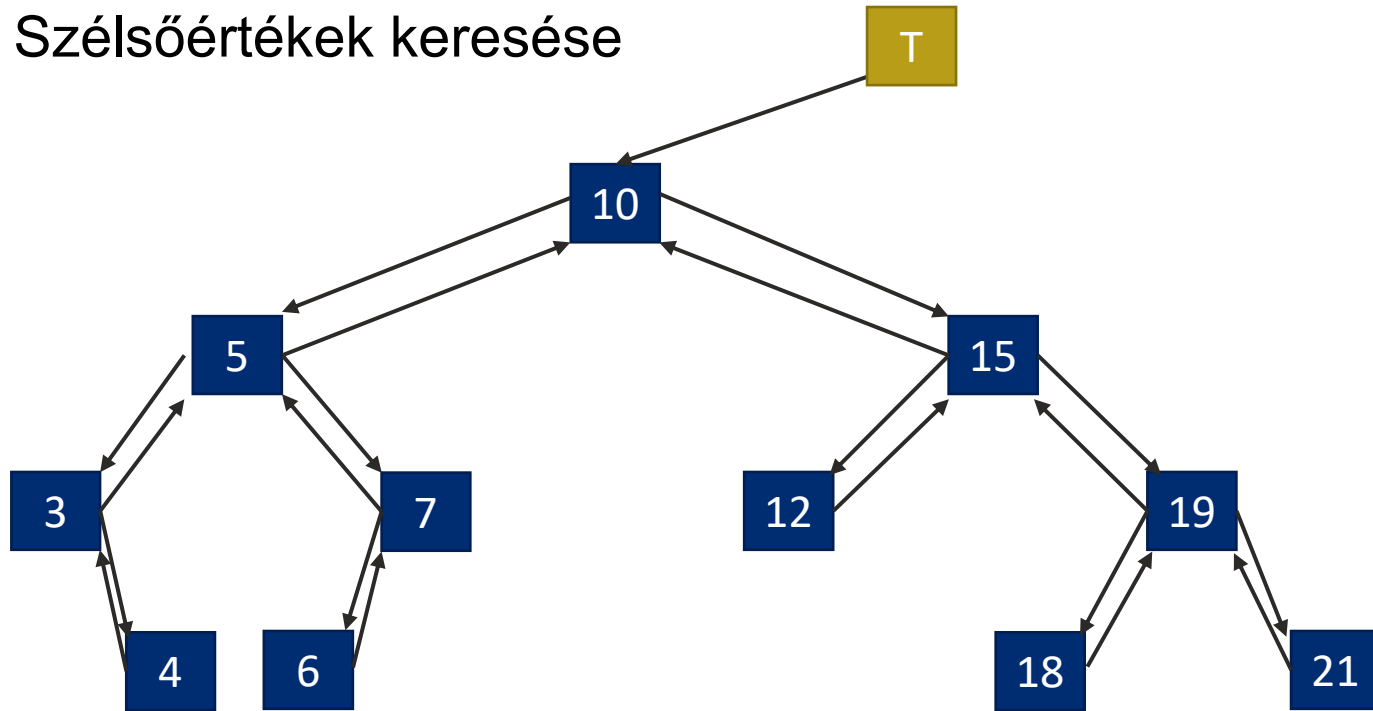
# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

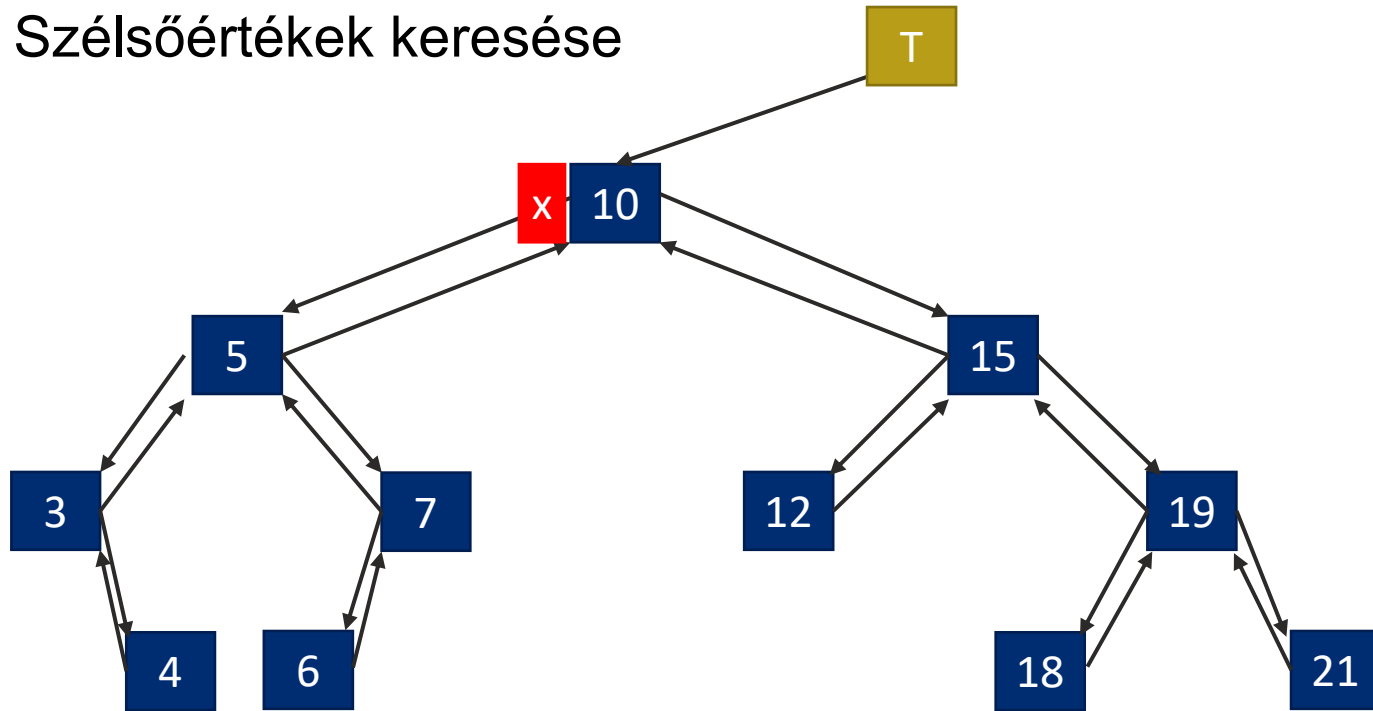
```
x ← gyökér[T]  
while bal[x] ≠ NIL  
do x ← bal[x]  
return x
```

Fában-maximum (T)

```
x ← gyökér[T]  
while jobb[x] ≠ NIL  
do x ← jobb[x]  
return x
```

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

$x \leftarrow \text{gyökér}[T]$

while  $\text{bal}[x] \neq \text{NIL}$

do  $x \leftarrow \text{bal}[x]$

return  $x$

Fában-maximum (T)

$x \leftarrow \text{gyökér}[T]$

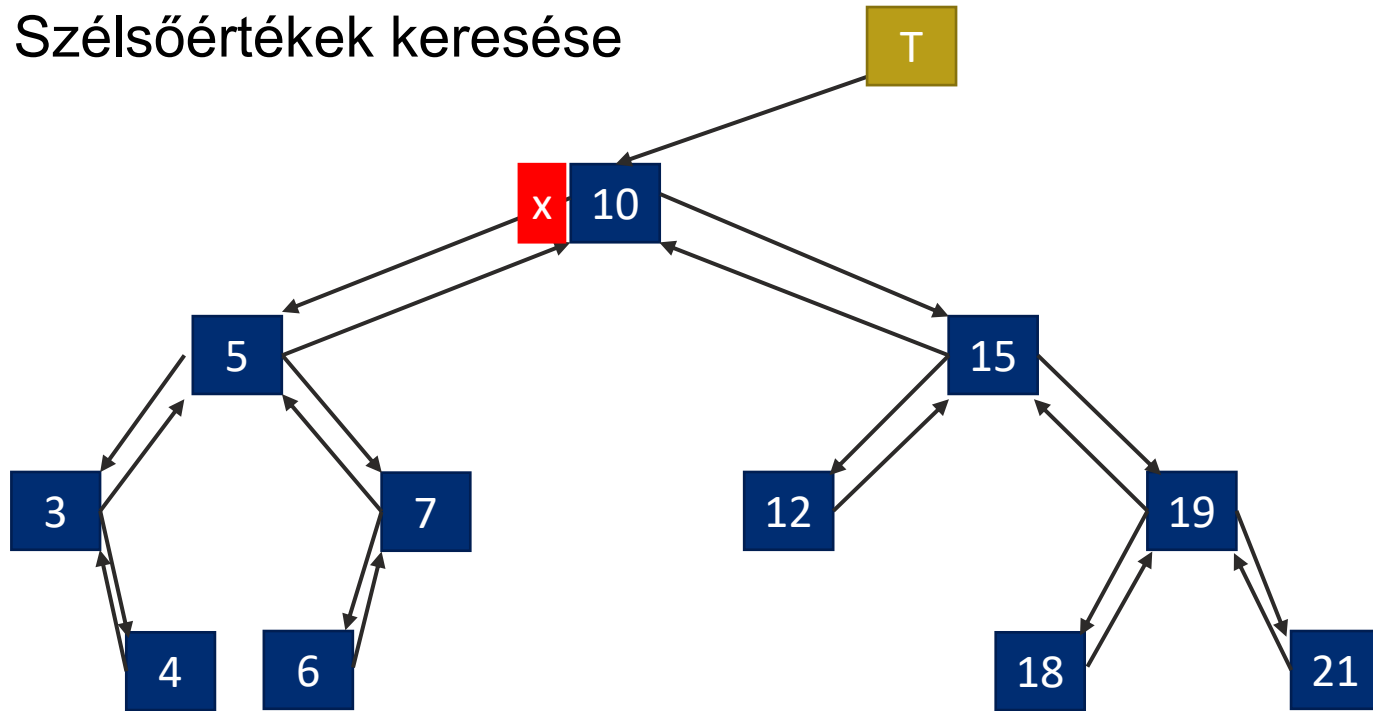
while  $\text{jobb}[x] \neq \text{NIL}$

do  $x \leftarrow \text{jobb}[x]$

return  $x$

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

$x \leftarrow \text{gyökér}[T]$

**while**  $\text{bal}[x] \neq \text{NIL}$

**do**  $x \leftarrow \text{bal}[x]$

**return**  $x$

Fában-maximum (T)

$x \leftarrow \text{gyökér}[T]$

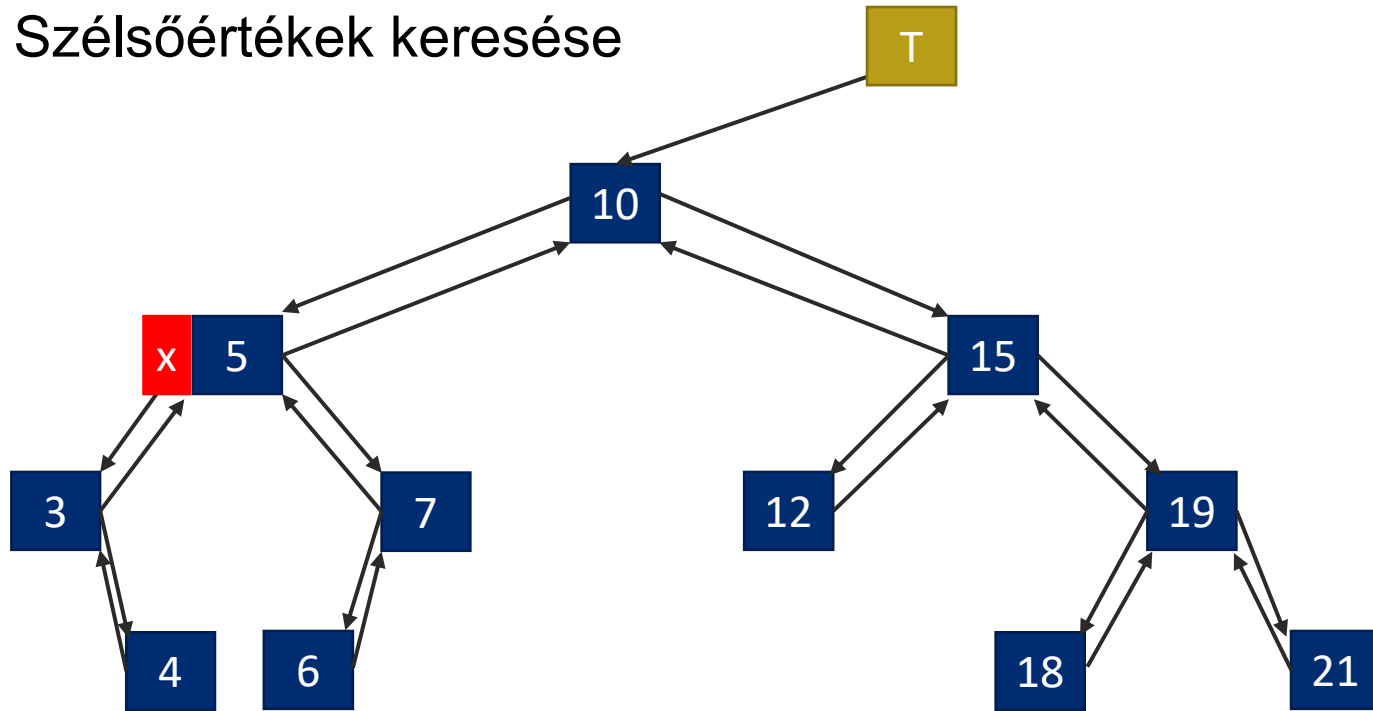
**while**  $\text{jobb}[x] \neq \text{NIL}$

**do**  $x \leftarrow \text{jobb}[x]$

**return**  $x$

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

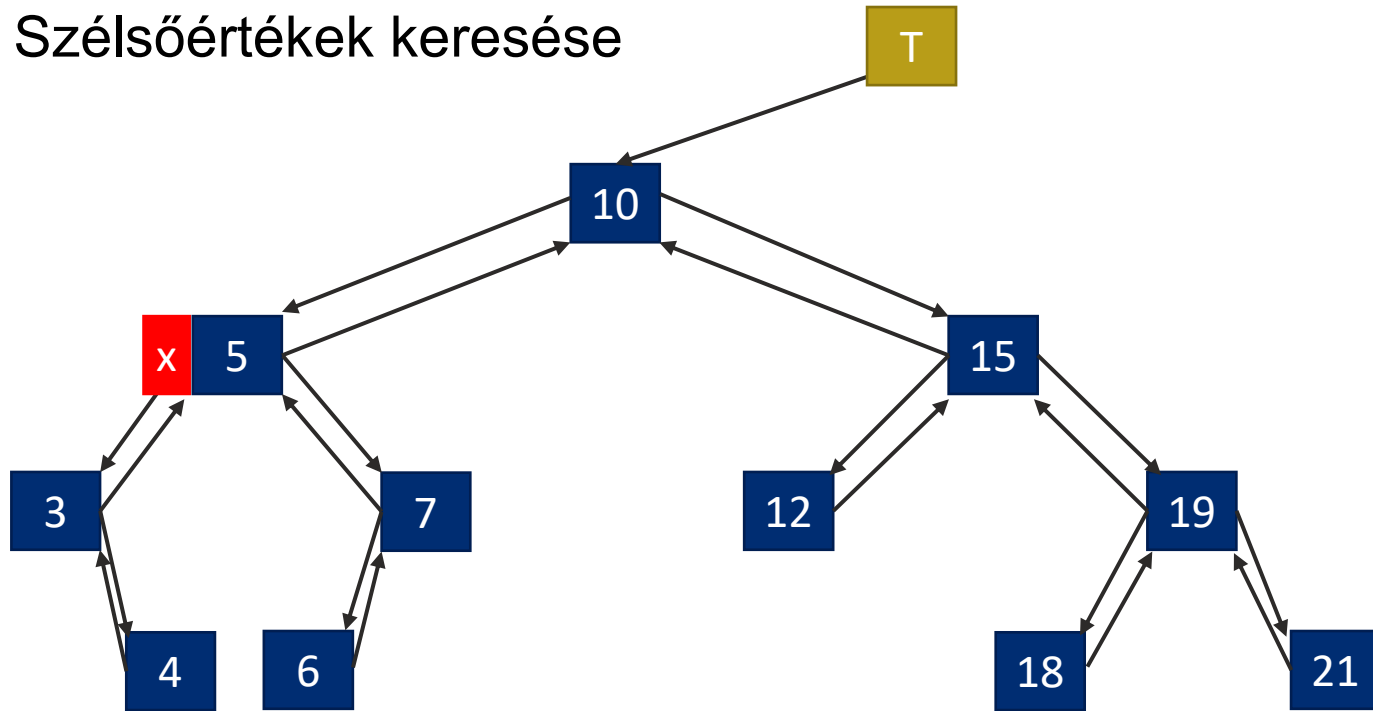
```
x ← gyökér[T]  
while bal[x] ≠ NIL  
do x ← bal[x]  
return x
```

Fában-maximum (T)

```
x ← gyökér[T]  
while jobb[x] ≠ NIL  
do x ← jobb[x]  
return x
```

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

$x \leftarrow \text{gyökér}[T]$

**while**  $\text{bal}[x] \neq \text{NIL}$

**do**  $x \leftarrow \text{bal}[x]$

**return**  $x$

Fában-maximum (T)

$x \leftarrow \text{gyökér}[T]$

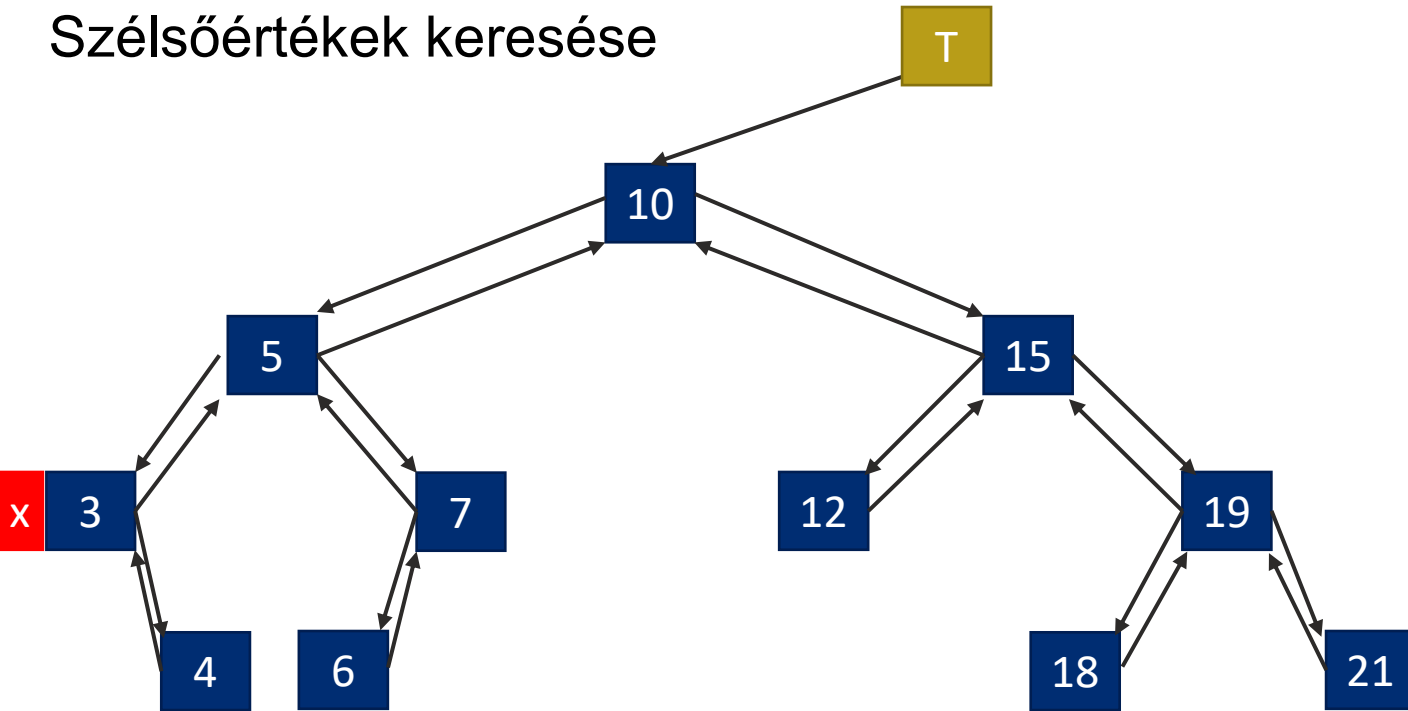
**while**  $\text{jobb}[x] \neq \text{NIL}$

**do**  $x \leftarrow \text{jobb}[x]$

**return**  $x$

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

```
x ← gyökér[T]  
while bal[x] ≠ NIL  
do x ← bal[x]  
return x
```

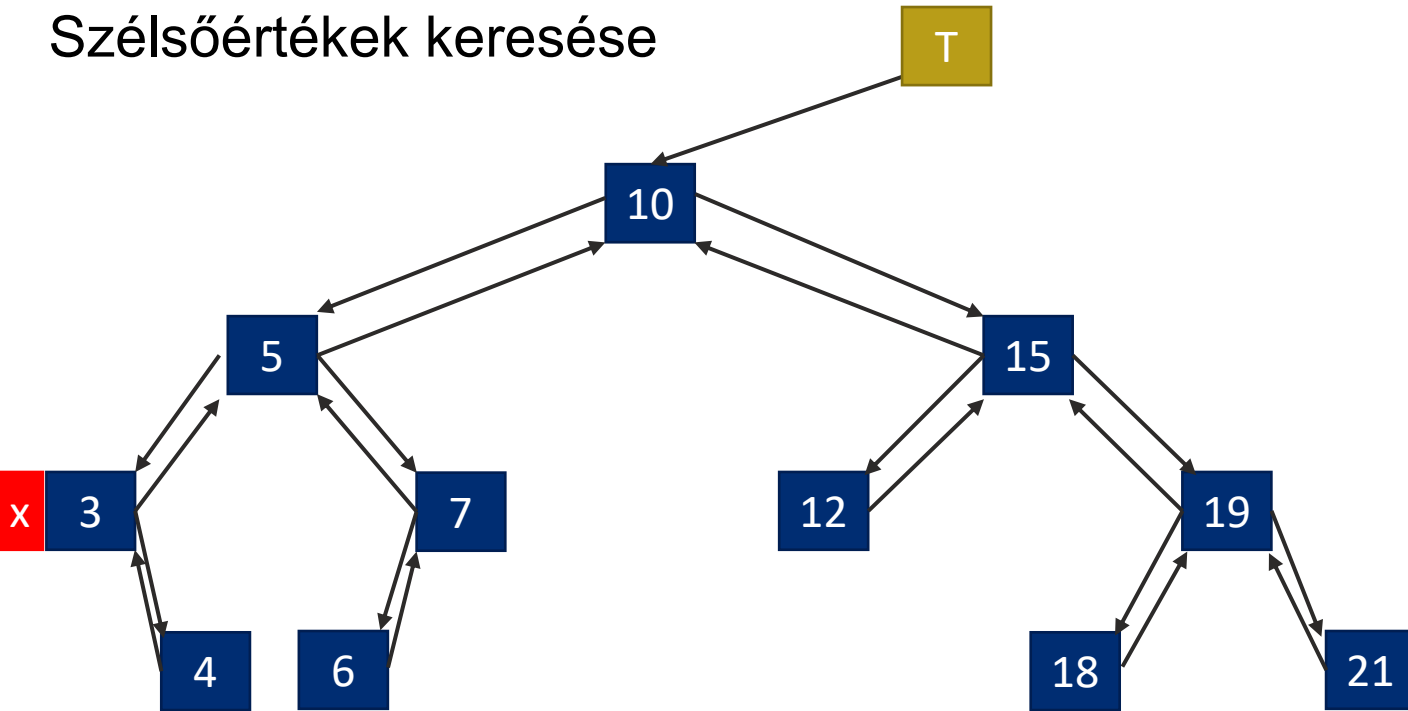
Fában-maximum (T)

```
x ← gyökér[T]  
while jobb[x] ≠ NIL  
do x ← jobb[x]  
return x
```



# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

$x \leftarrow \text{gyökér}[T]$

**while**  $\text{bal}[x] \neq \text{NIL}$

do  $x \leftarrow \text{bal}[x]$

return  $x$

Fában-maximum (T)

$x \leftarrow \text{gyökér}[T]$

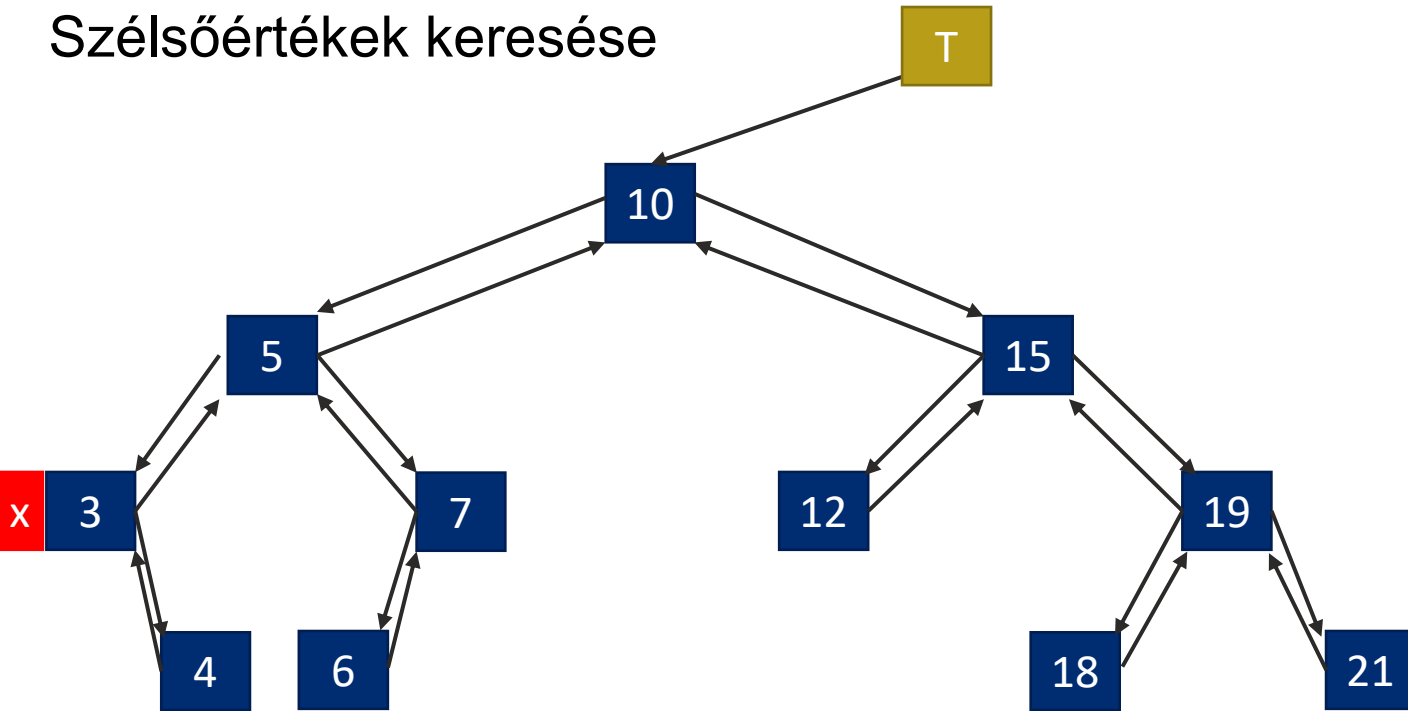
**while**  $\text{jobb}[x] \neq \text{NIL}$

do  $x \leftarrow \text{jobb}[x]$

return  $x$

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

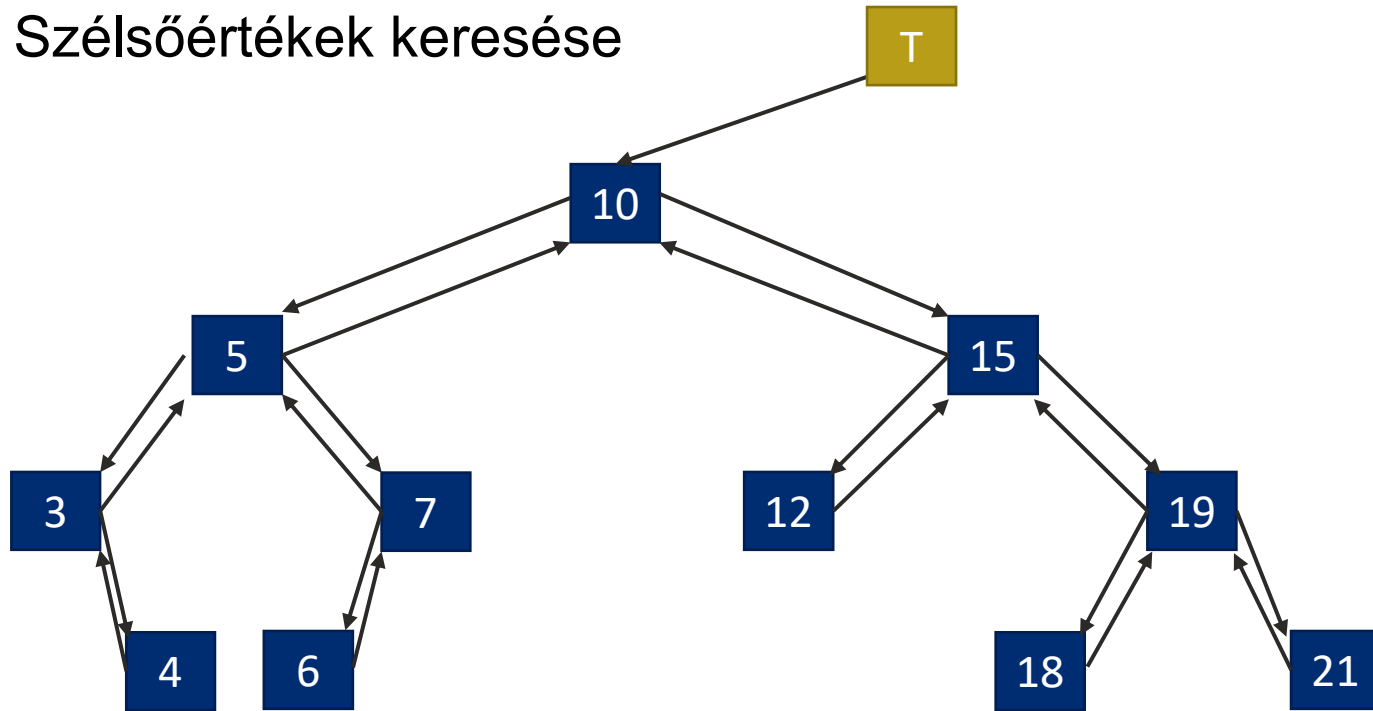
```
x ← gyökér[T]  
while bal[x] ≠ NIL  
do x ← bal[x]  
return x
```

Fában-maximum (T)

```
x ← gyökér[T]  
while jobb[x] ≠ NIL  
do x ← jobb[x]  
return x
```

# Bináris keresési fa – Szélsőértékek

Szélsőértékek keresése



Fában-minimum (T)

```
x ← gyökér[T]  
while bal[x] ≠ NIL  
do x ← bal[x]  
return x
```

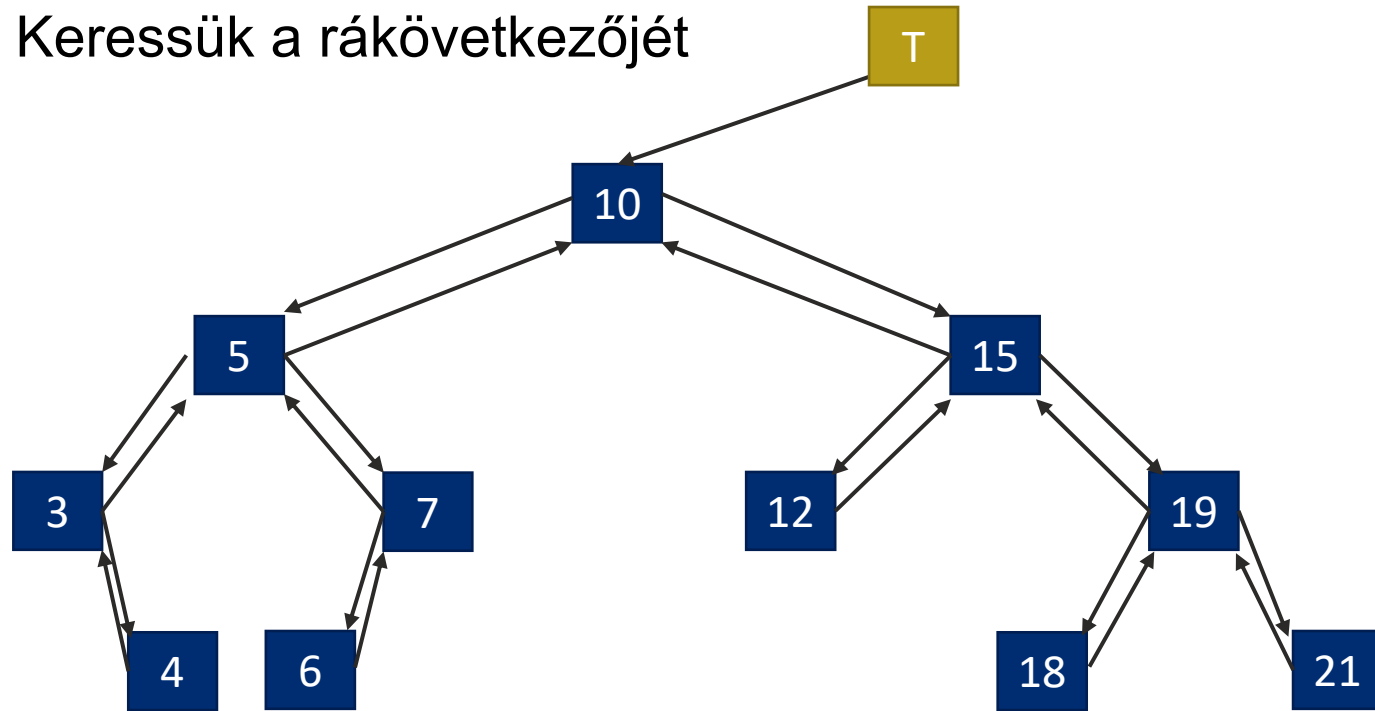
Fában-maximum (T)

```
x ← gyökér[T]  
while jobb[x] ≠ NIL  
do x ← jobb[x]  
return x
```

Maximum keresése analóg módon történik.

# Bináris keresési fa – Rákövetkező elem

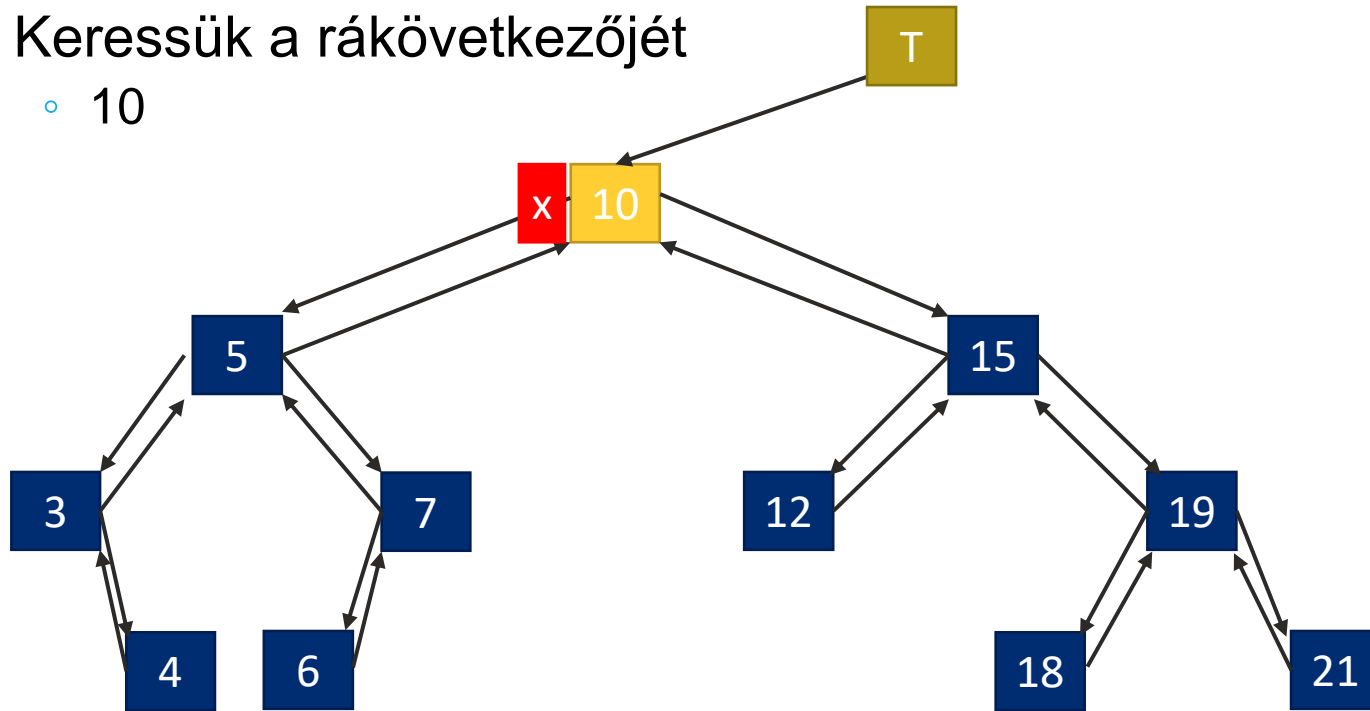
Keressük a rákövetkezőjét



# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

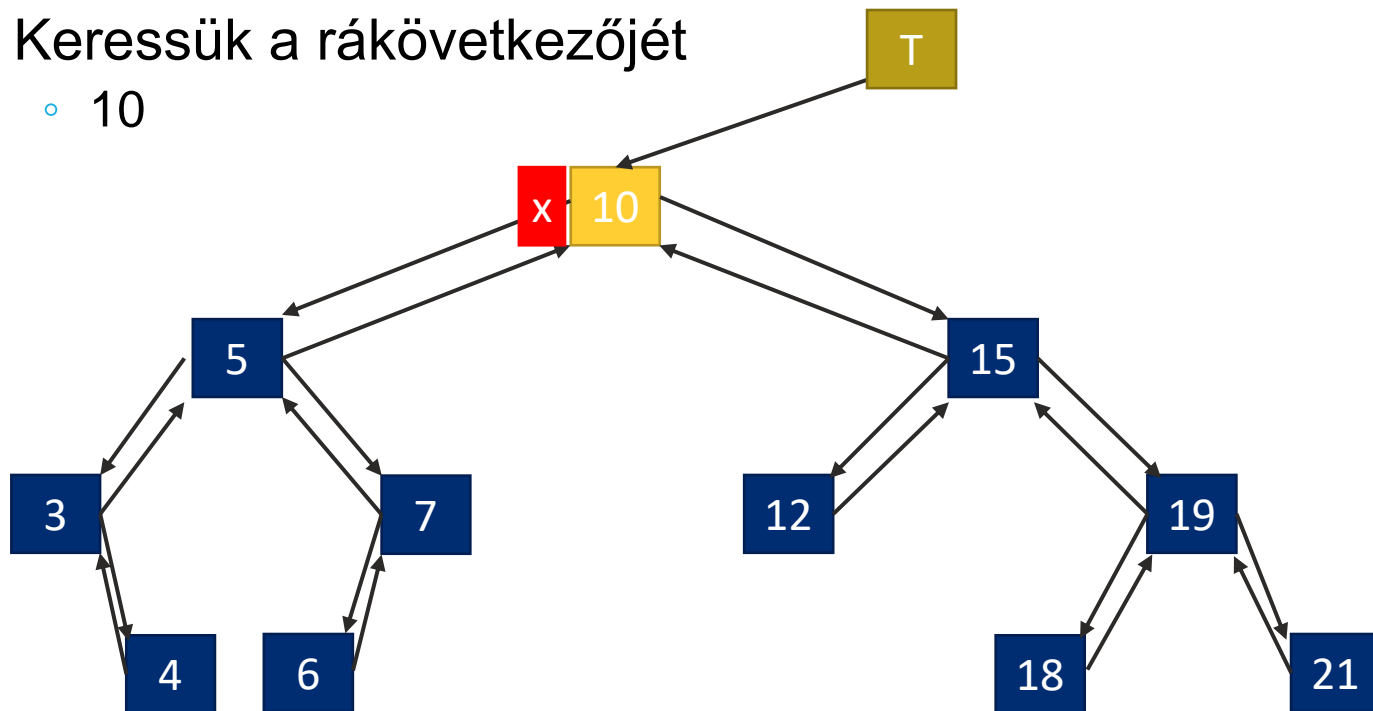
- 10



```
Fában-következő(T, x)
if jobb[x] ≠ NIL
    then return Fában-minimum (jobb[x])
y ← szülő[x]
while y ≠ NIL és x = jobb[y] do
    x ← y
    y ← szülő[x]
return y
```

Keressük a rákövetkezőjét

- 10



Fában-következő( $T, x$ )

if jobb[x]  $\neq$  NIL

then return Fában-minimum (jobb[x])

y  $\leftarrow$  szülő[x]

while y  $\neq$  NIL és x = jobb[y] do

x  $\leftarrow$  y

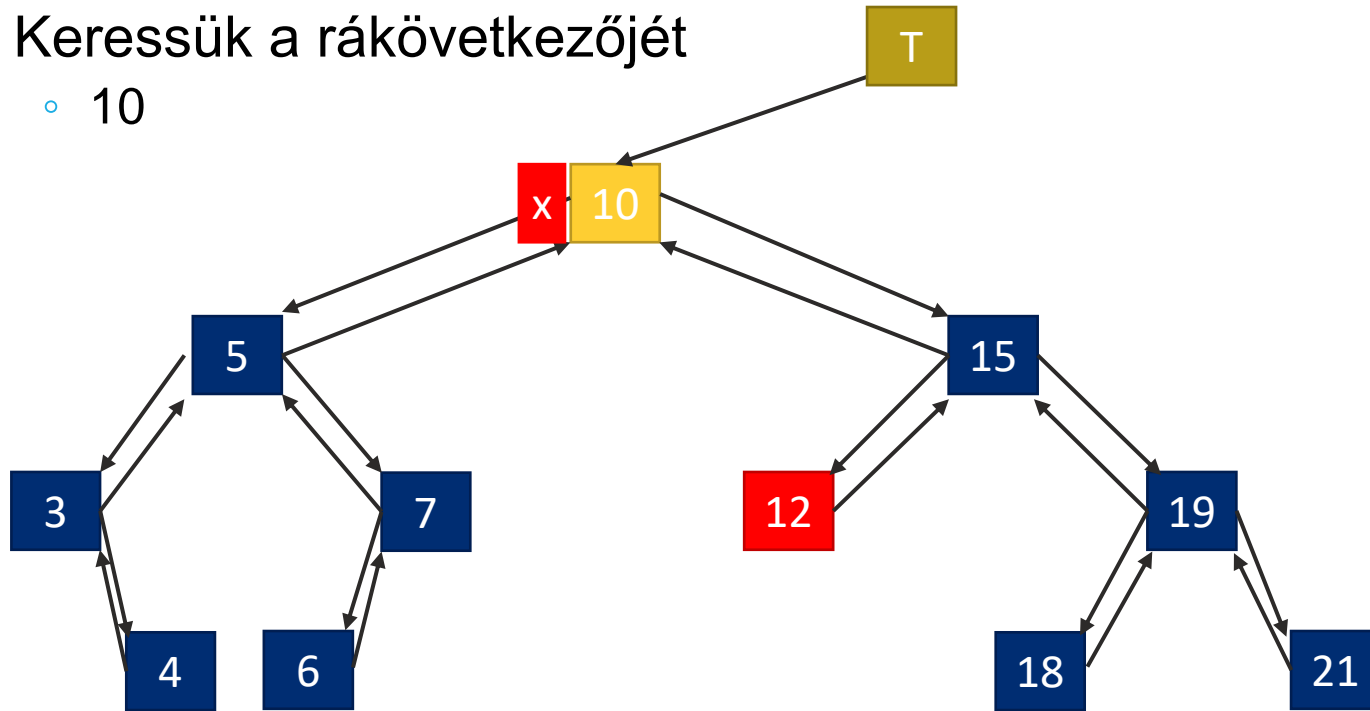
y  $\leftarrow$  szülő[x]

return y

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 10



Fában-következő( $T, x$ )

```
if jobb[x]  $\neq$  NIL  
    then return Fában-minimum (jobb[x])
```

```
y  $\leftarrow$  szülő[x]
```

```
while y  $\neq$  NIL és x = jobb[y] do
```

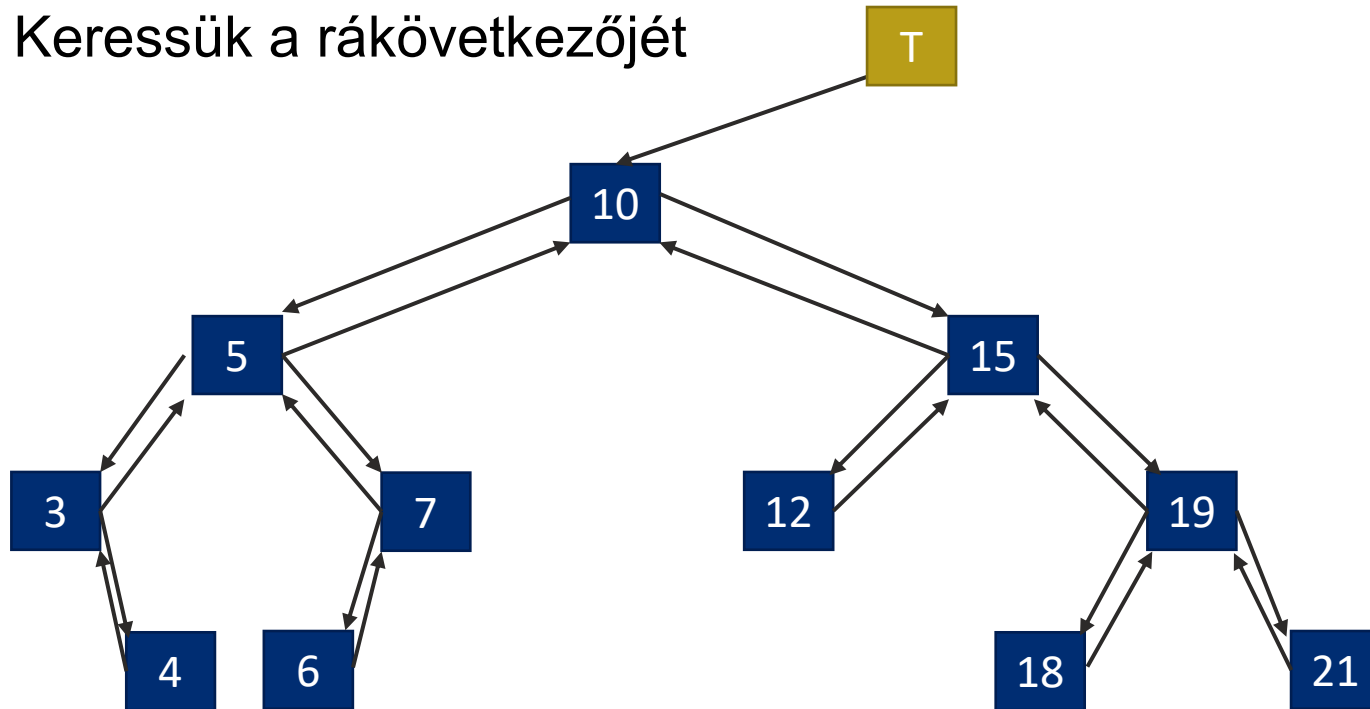
```
    x  $\leftarrow$  y
```

```
    y  $\leftarrow$  szülő[x]
```

```
return y
```

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét



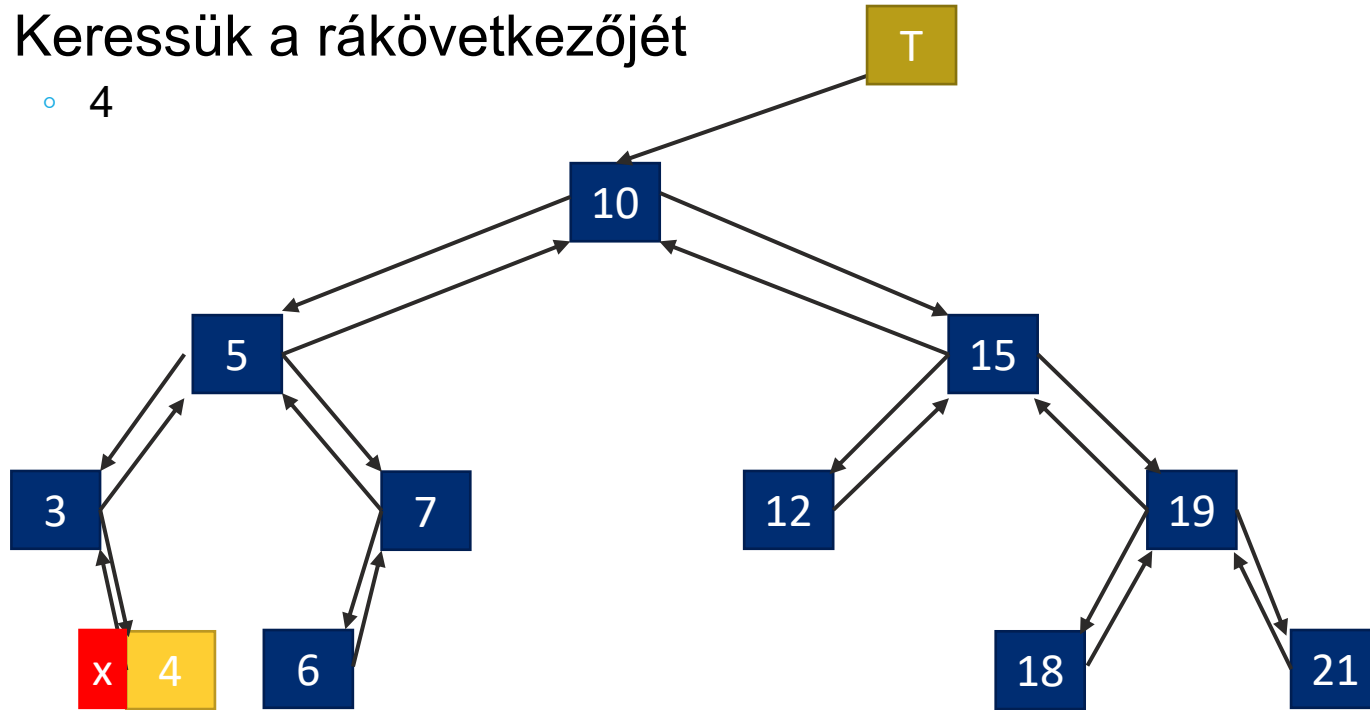
```
Fában-következő(T, x)
if jobb[x] ≠ NIL
    then return Fában-minimum (jobb[x])
y ← szülő[x]
while y ≠ NIL és x = jobb[y] do
    x ← y
    y ← szülő[x]
return y
```



# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 4

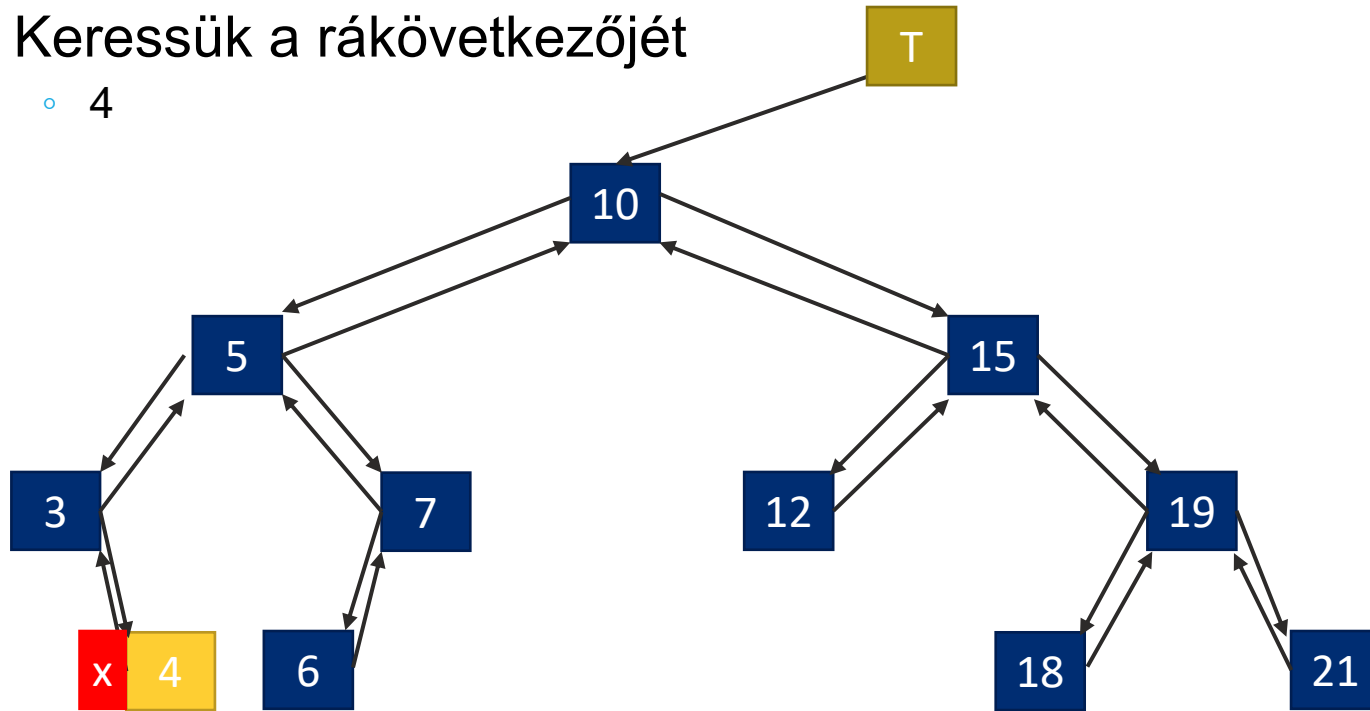


```
Fában-következő(T, x)
if jobb[x] ≠ NIL
    then return Fában-minimum (jobb[x])
y ← szülő[x]
while y ≠ NIL és x = jobb[y] do
    x ← y
    y ← szülő[x]
return y
```

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

◦ 4



Fában-következő( $T, x$ )

**if** jobb[x]  $\neq$  NIL

then return Fában-minimum (jobb[x])

$y \leftarrow$  szülő[x]

while  $y \neq$  NIL és  $x =$  jobb[y] do

$x \leftarrow y$

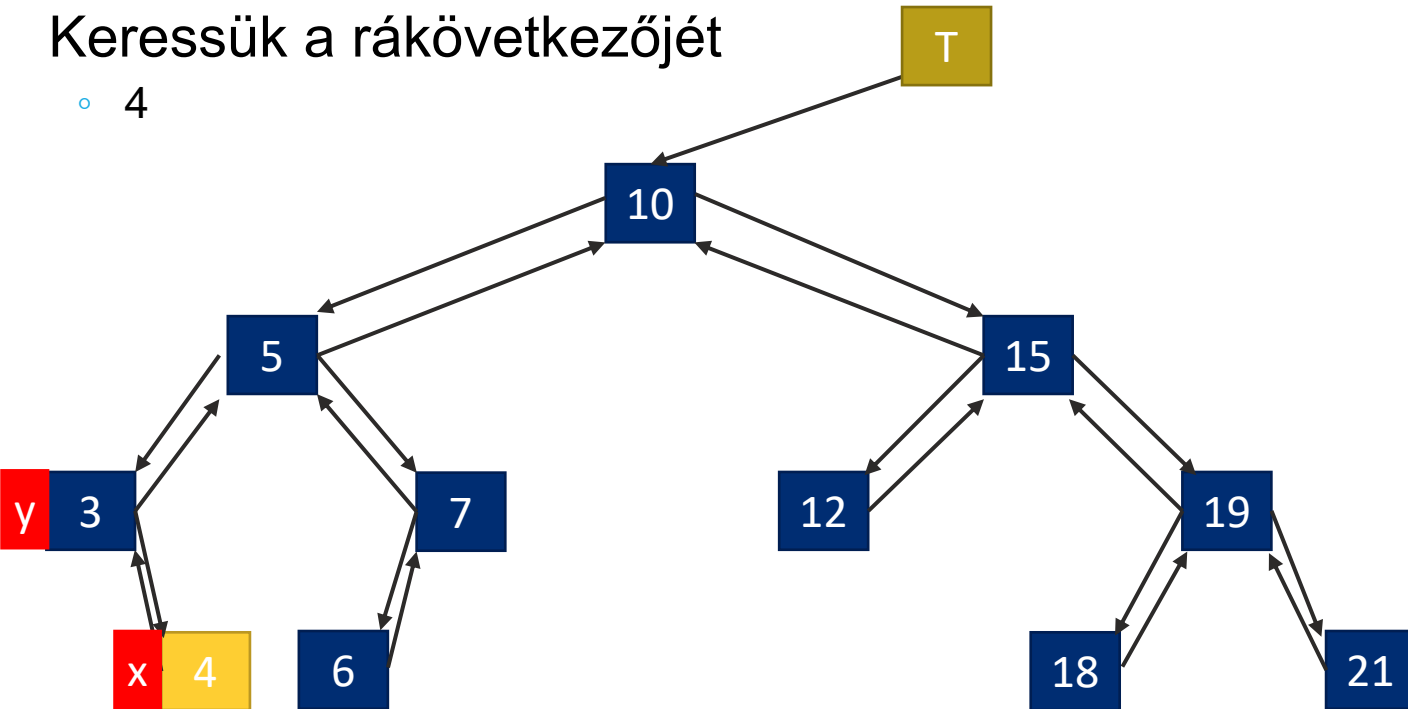
$y \leftarrow$  szülő[x]

return y

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 4



Fában-következő( $T, x$ )

if jobb[x]  $\neq$  NIL

then return Fában-minimum (jobb[x])

$y \leftarrow$  szülő[x]

while  $y \neq$  NIL és  $x =$  jobb[y] do

$x \leftarrow y$

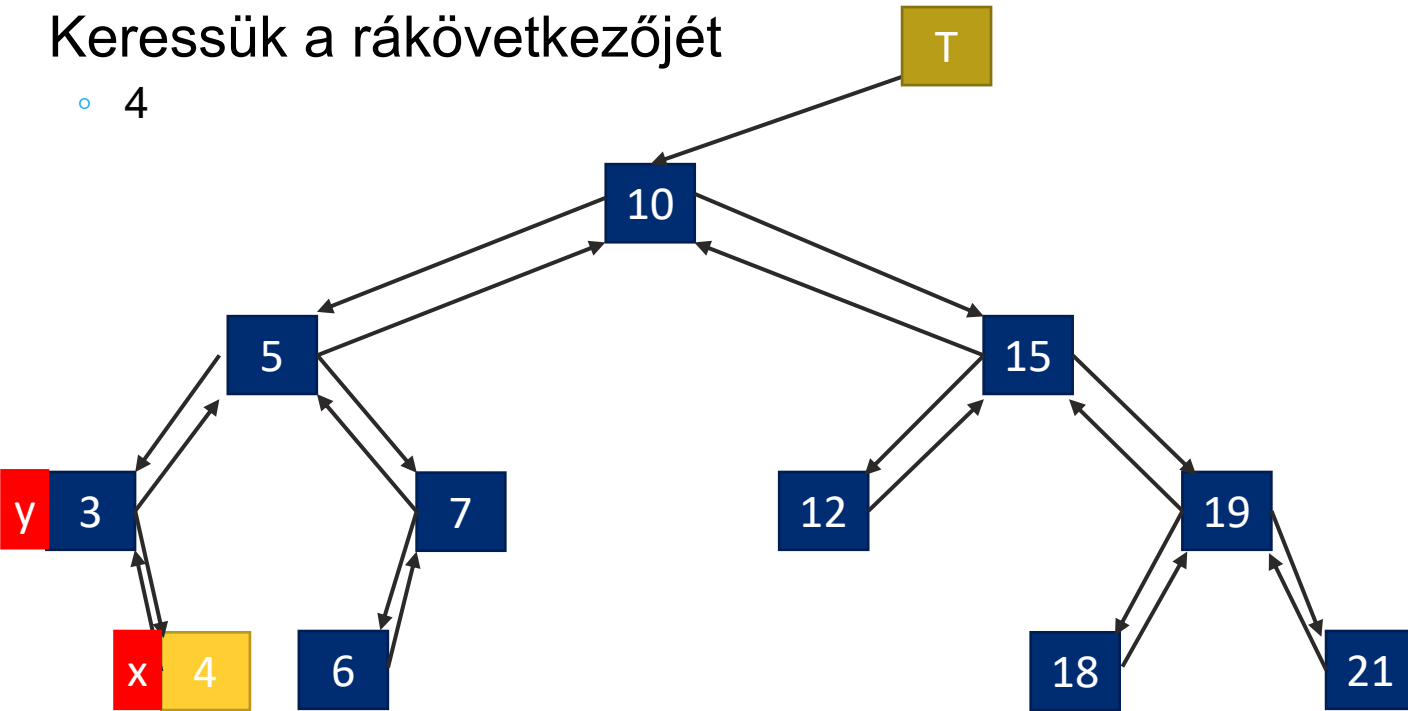
$y \leftarrow$  szülő[x]

return y

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 4



Fában-következő( $T, x$ )

if jobb[x]  $\neq$  NIL

then return Fában-minimum (jobb[x])

$y \leftarrow$  szülő[x]

while  $y \neq$  NIL és  $x =$  jobb[y] do

$x \leftarrow y$

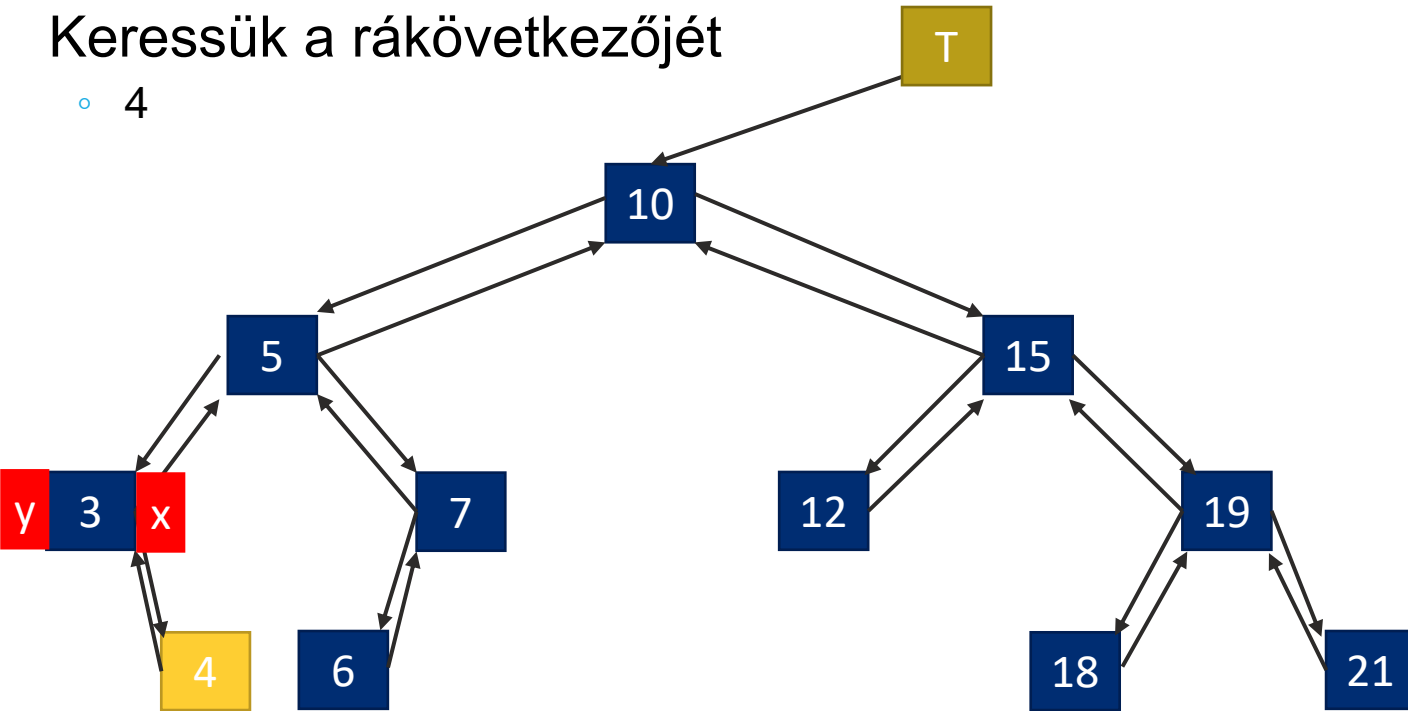
$y \leftarrow$  szülő[x]

return y

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

◦ 4



Fában-következő( $T, x$ )

if jobb[x]  $\neq$  NIL

then return Fában-minimum (jobb[x])

$y \leftarrow$  szülő[x]

while  $y \neq$  NIL és  $x =$  jobb[y] do

$x \leftarrow y$

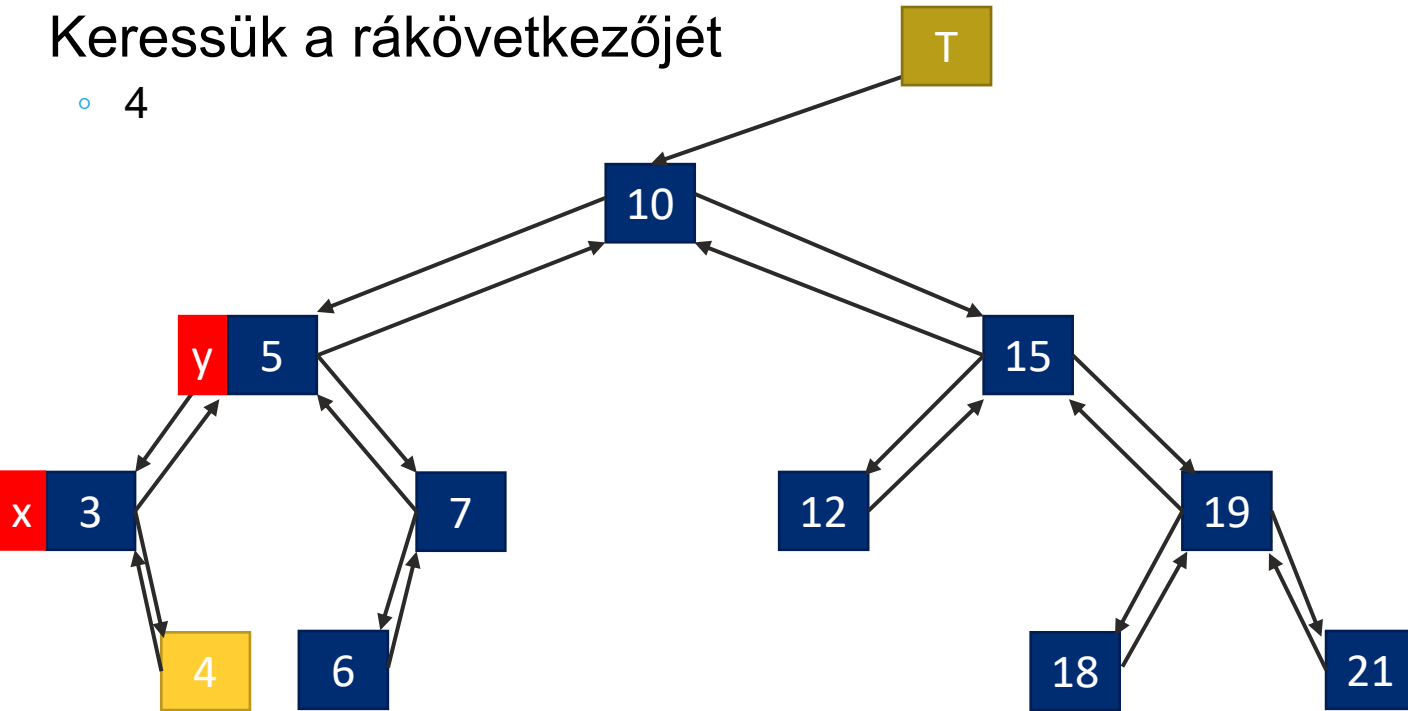
$y \leftarrow$  szülő[x]

return y

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 4



Fában-következő( $T, x$ )

if jobb[x]  $\neq$  NIL

then return Fában-minimum (jobb[x])

$y \leftarrow$  szülő[x]

while  $y \neq$  NIL és  $x =$  jobb[y] do

$x \leftarrow y$

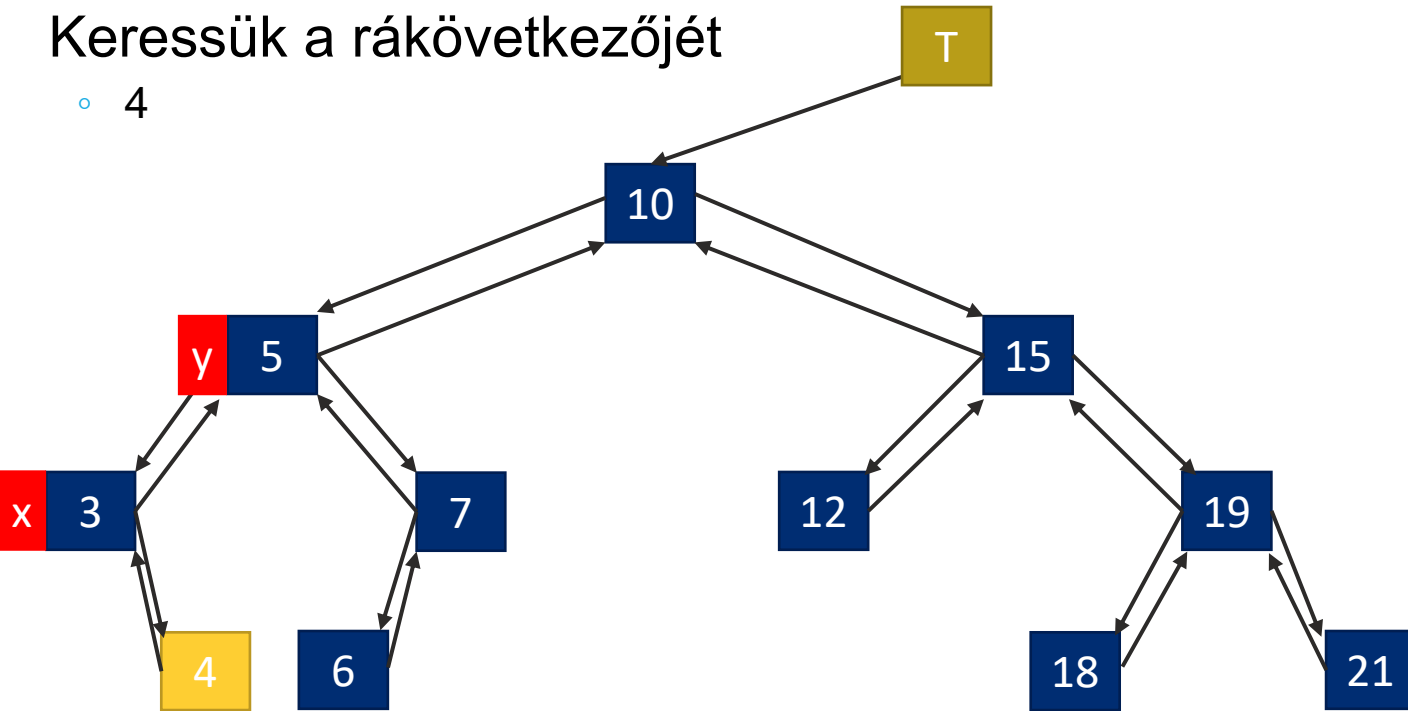
$y \leftarrow$  szülő[x]

return y

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 4

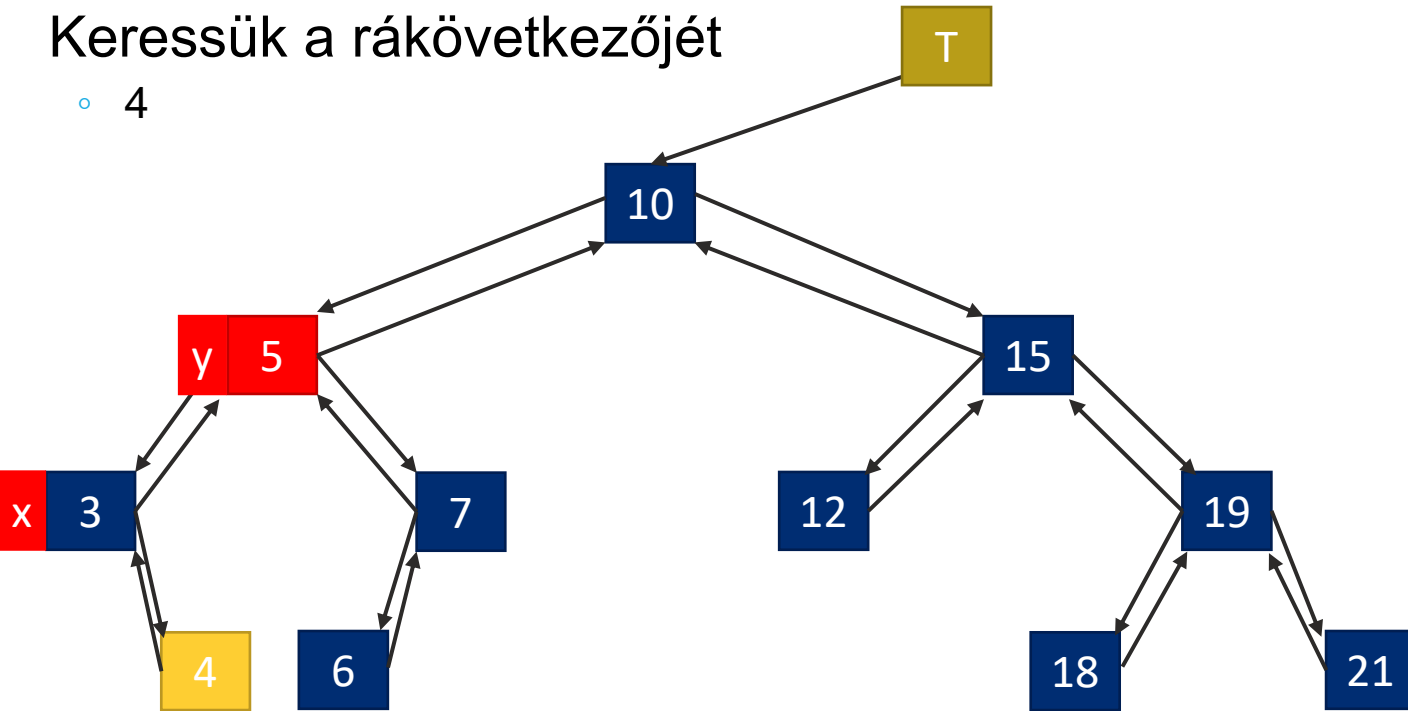


```
Fában-következő(T, x)
if jobb[x] ≠ NIL
    then return Fában-minimum (jobb[x])
y ← szülő[x]
while y ≠ NIL és x = jobb[y] do
    x ← y
    y ← szülő[x]
return y
```

# Bináris keresési fa – Rákövetkező elem

Keressük a rákövetkezőjét

- 4

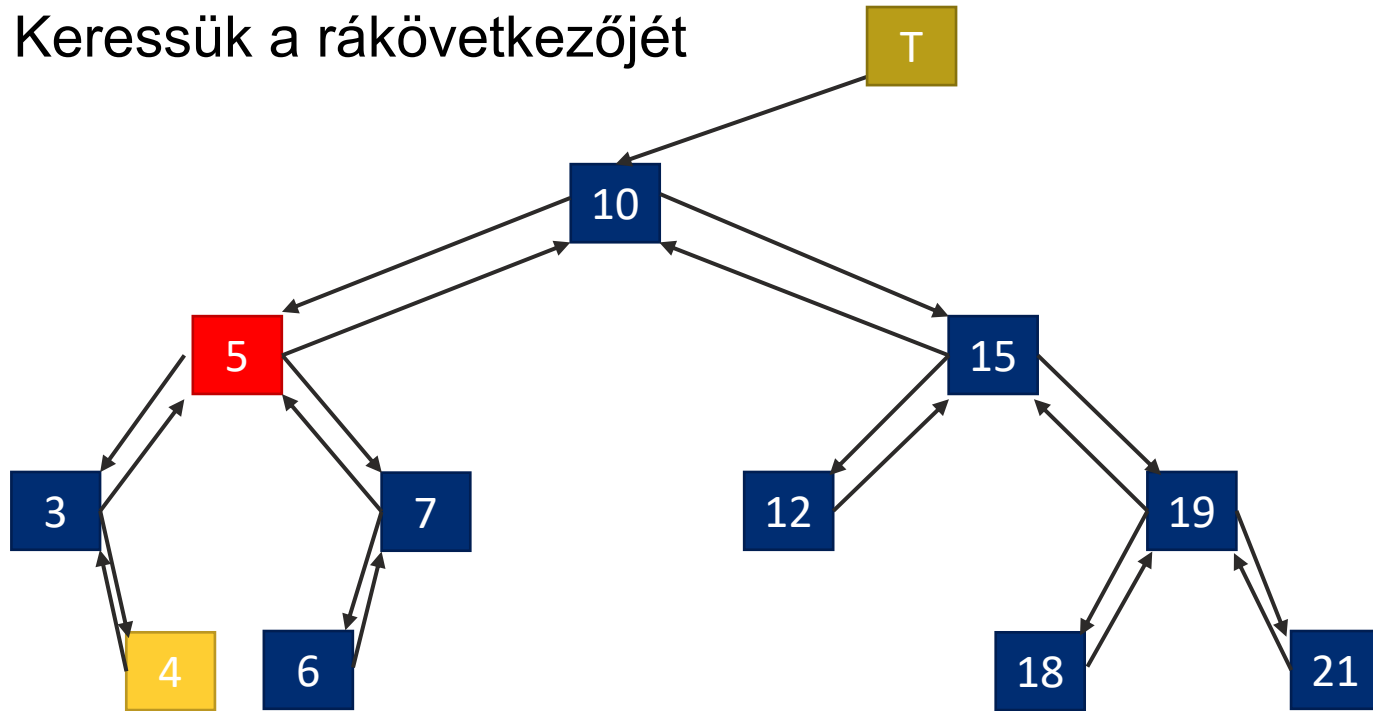


```
Fában-következő(T, x)
if jobb[x] ≠ NIL
    then return Fában-minimum (jobb[x])
y ← szülő[x]
while y ≠ NIL és x = jobb[y] do
    x ← y
    y ← szülő[x]
return y
```



# Bináris keresési fa – Rákövetkező elem

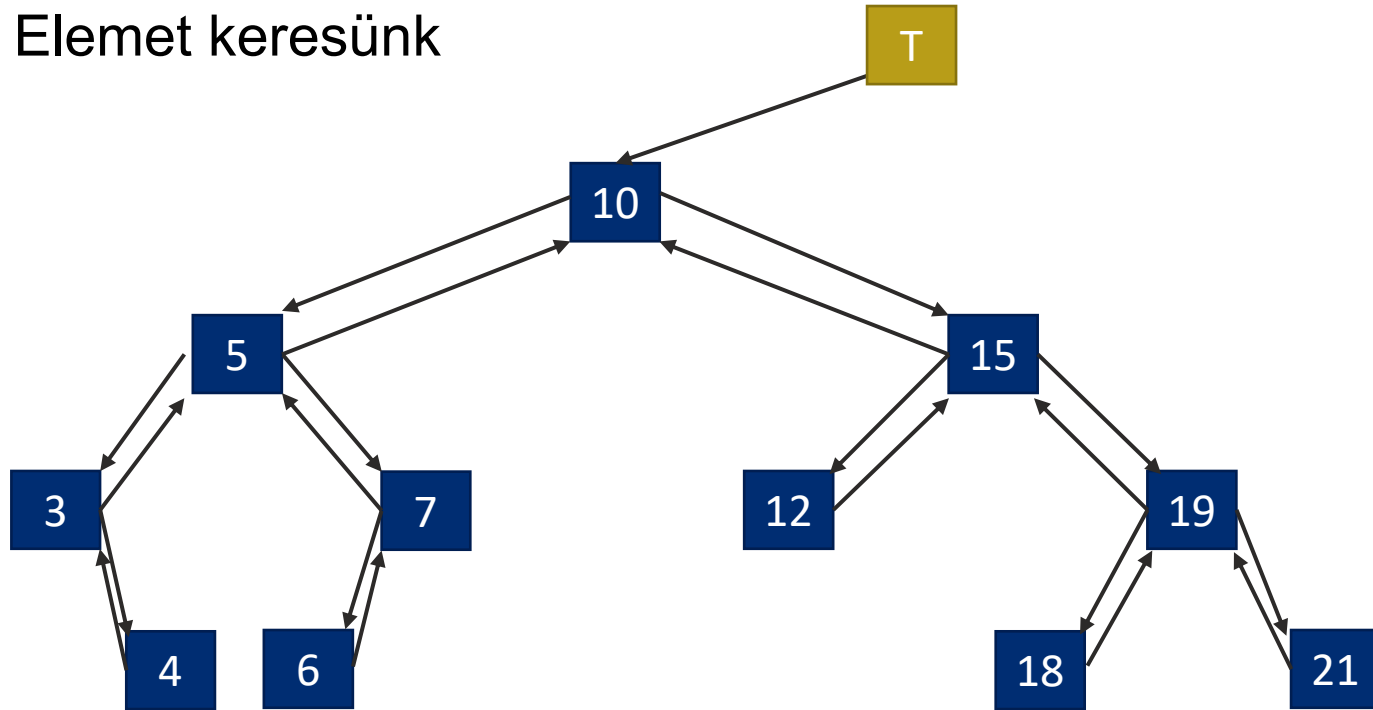
Keressük a rákövetkezőjét



```
Fában-következő(T, x)
if jobb[x] ≠ NIL
    then return Fában-minimum (jobb[x])
y ← szülő[x]
while y ≠ NIL és x = jobb[y] do
    x ← y
    y ← szülő[x]
return y
```

# Bináris keresési fa – Elem keresése

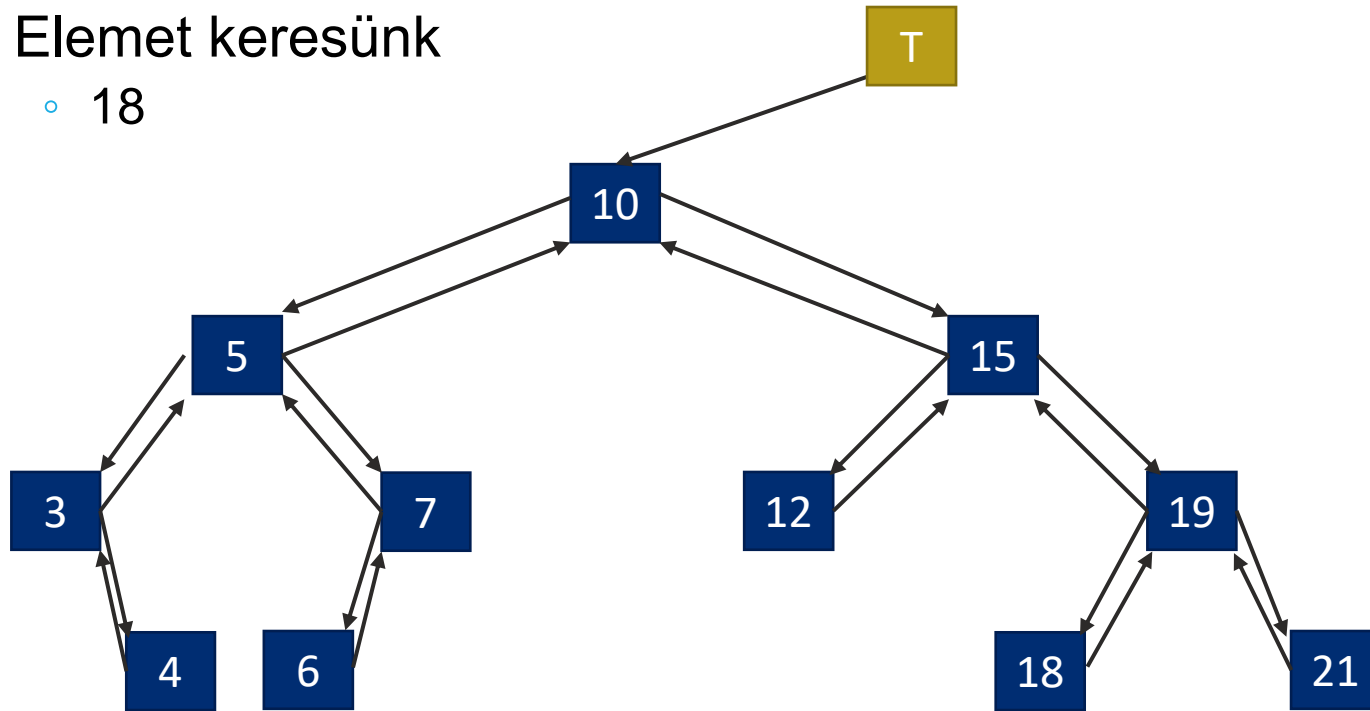
Elemet keresünk



# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

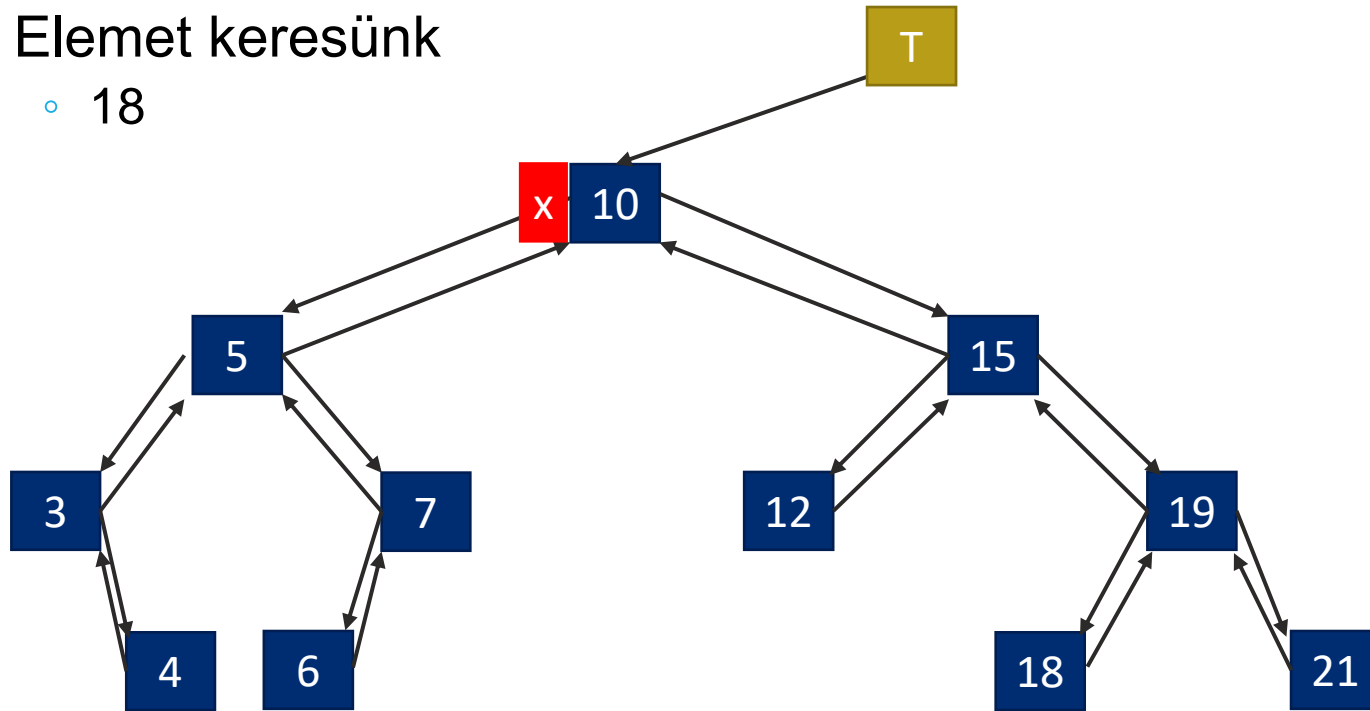


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

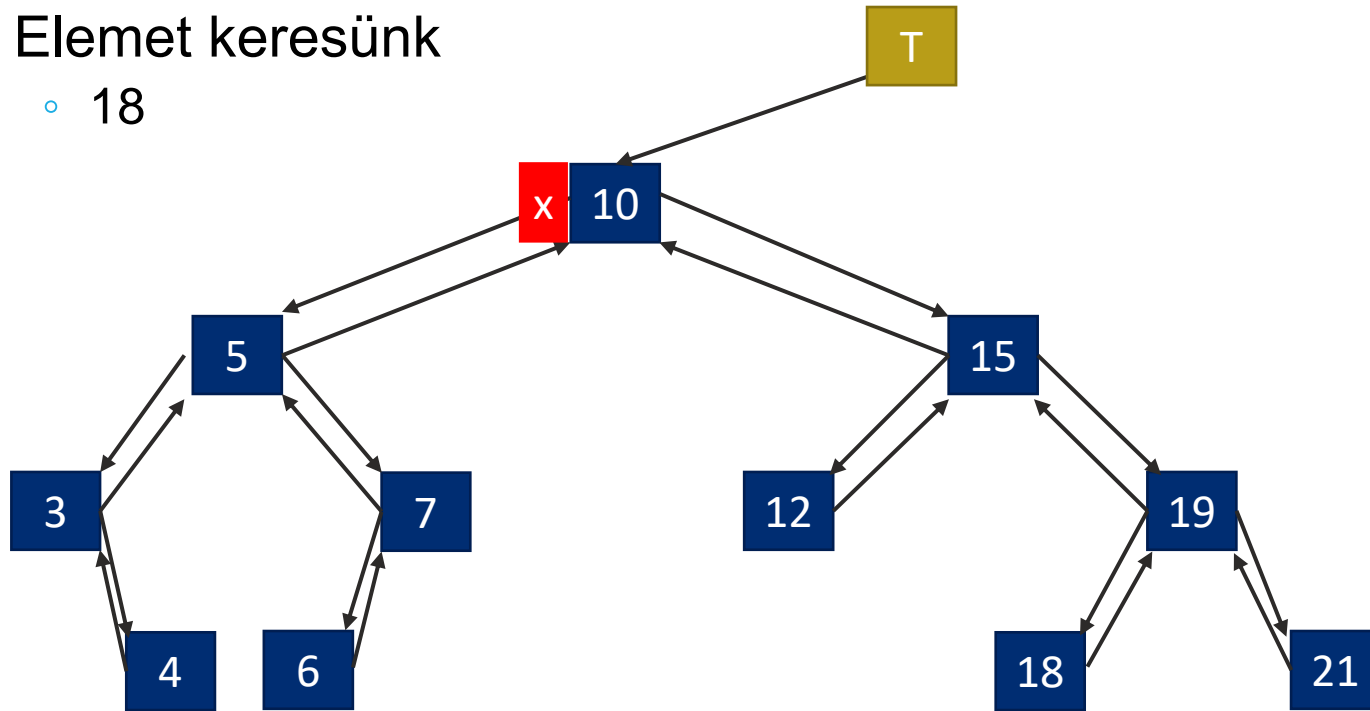


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

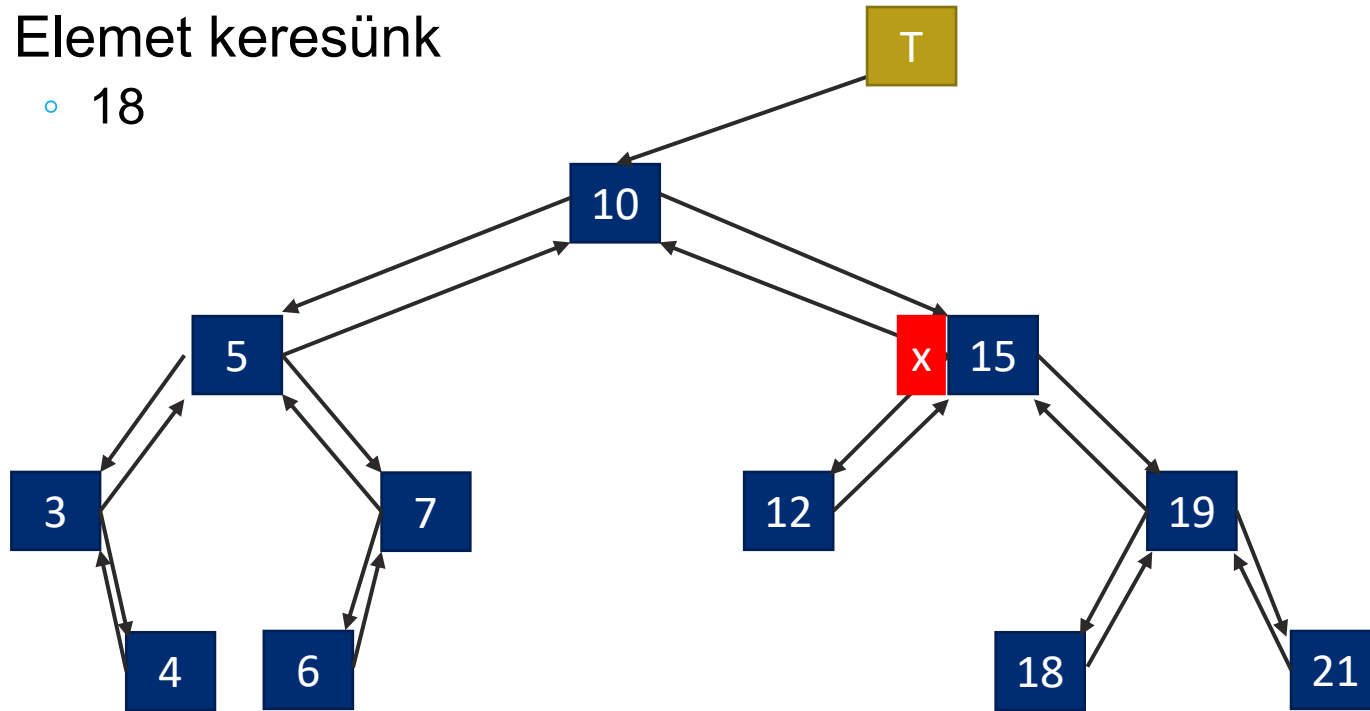


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
        else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

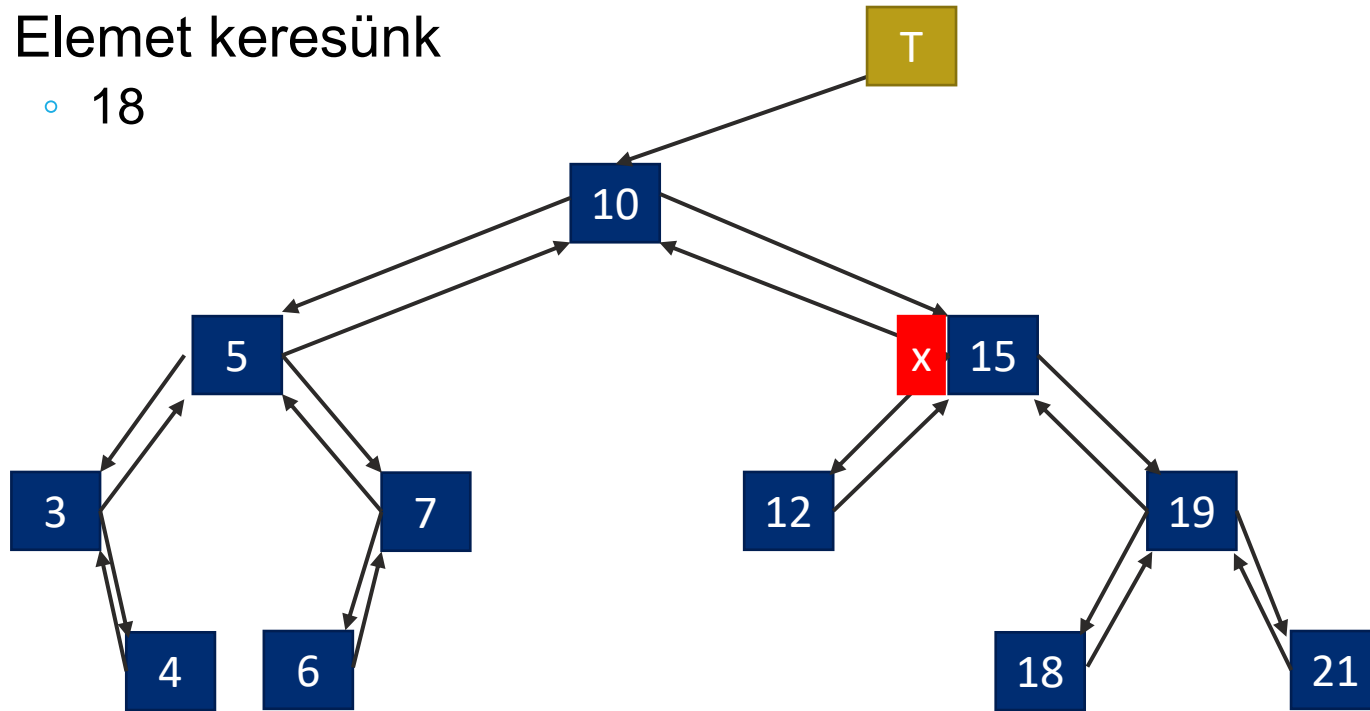


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

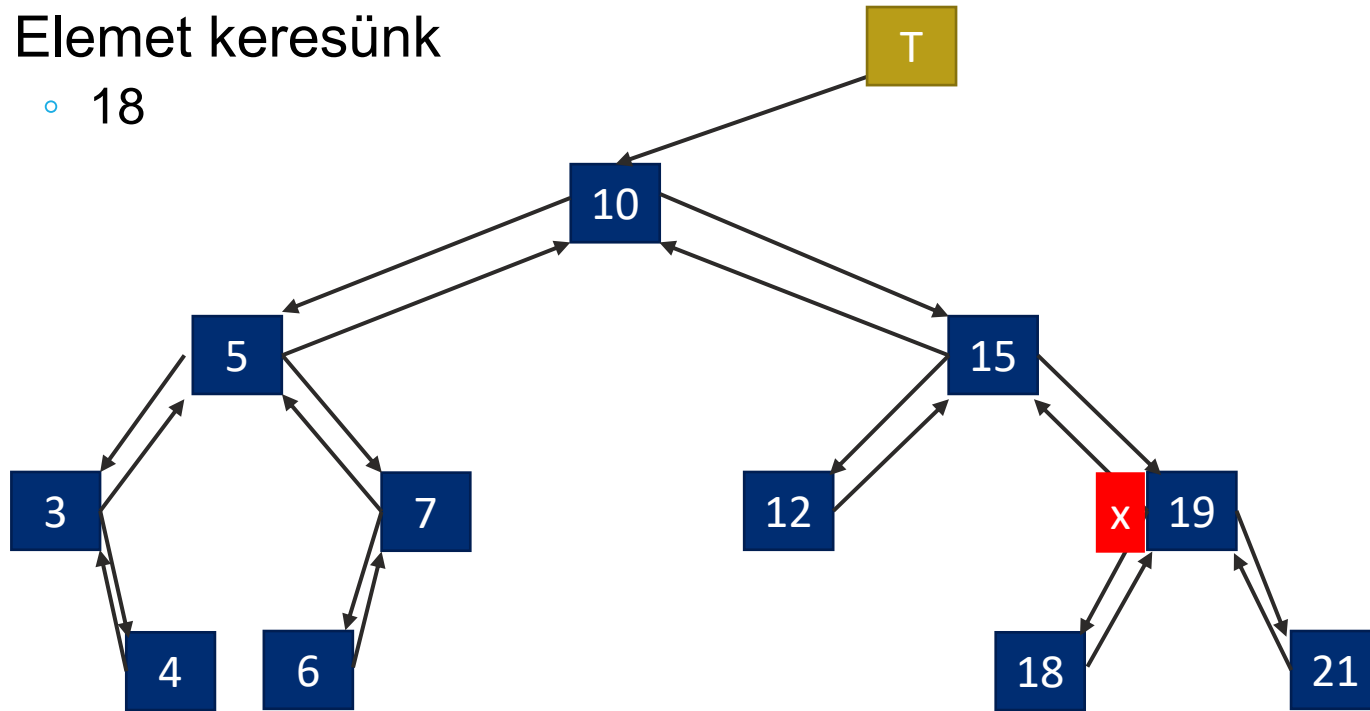


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18



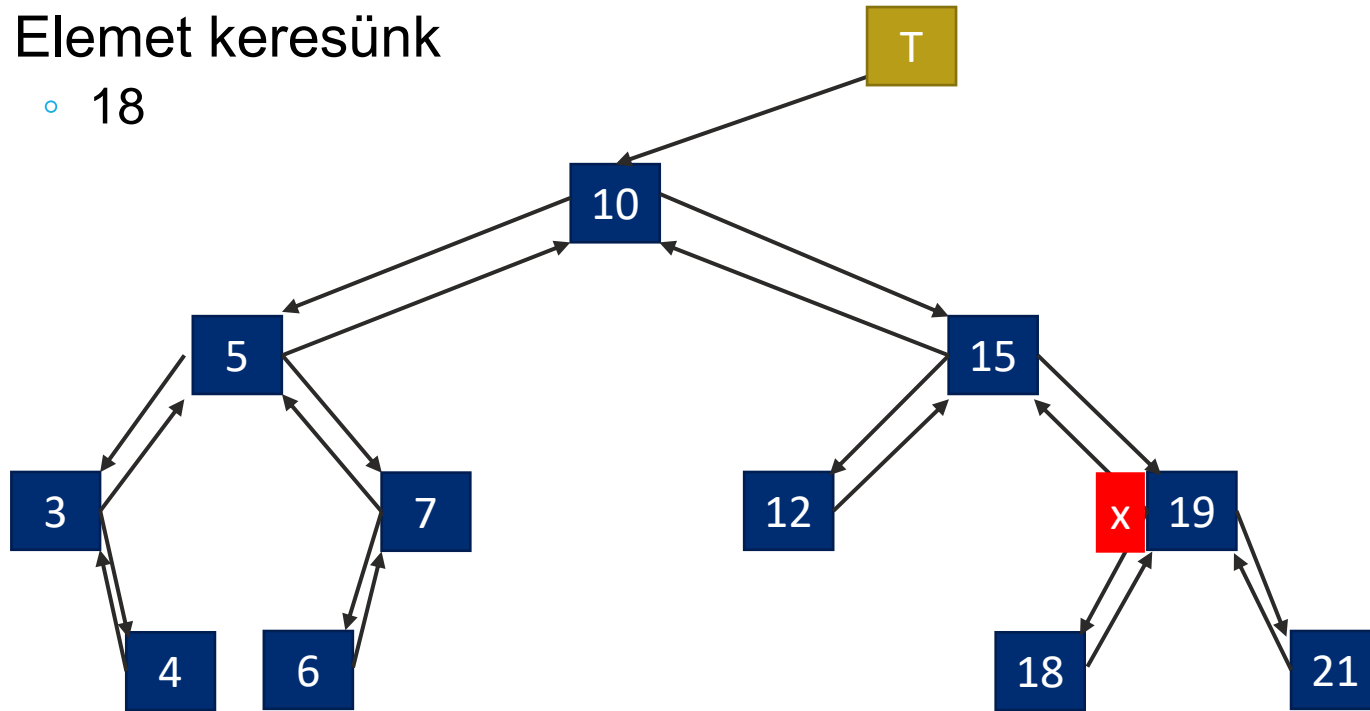
```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```



# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

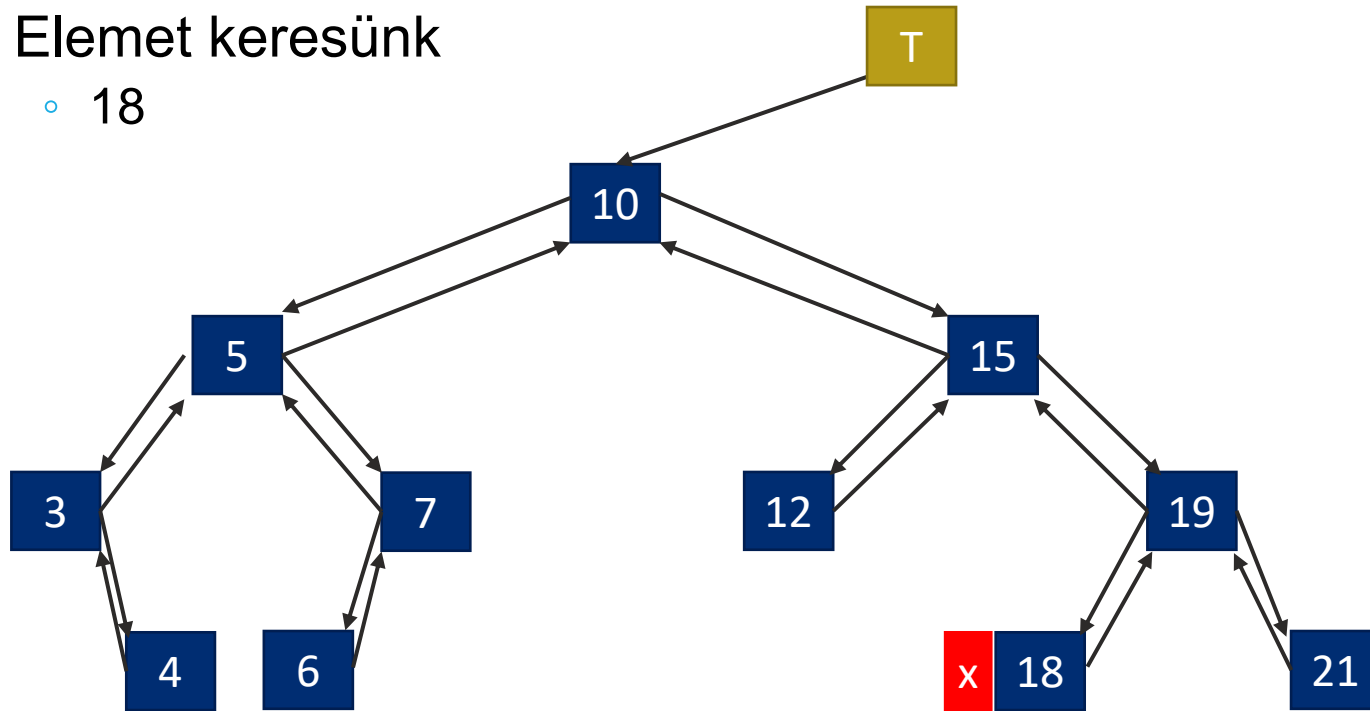


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
        else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

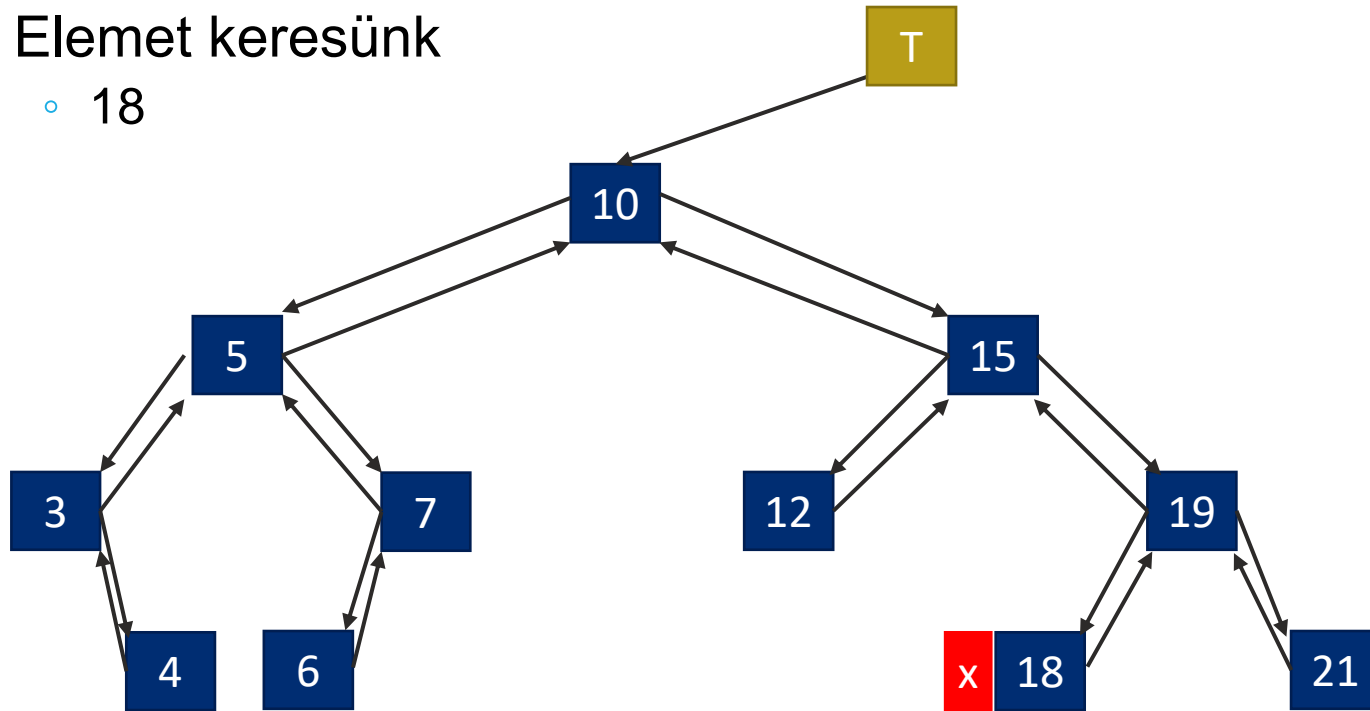


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
    then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

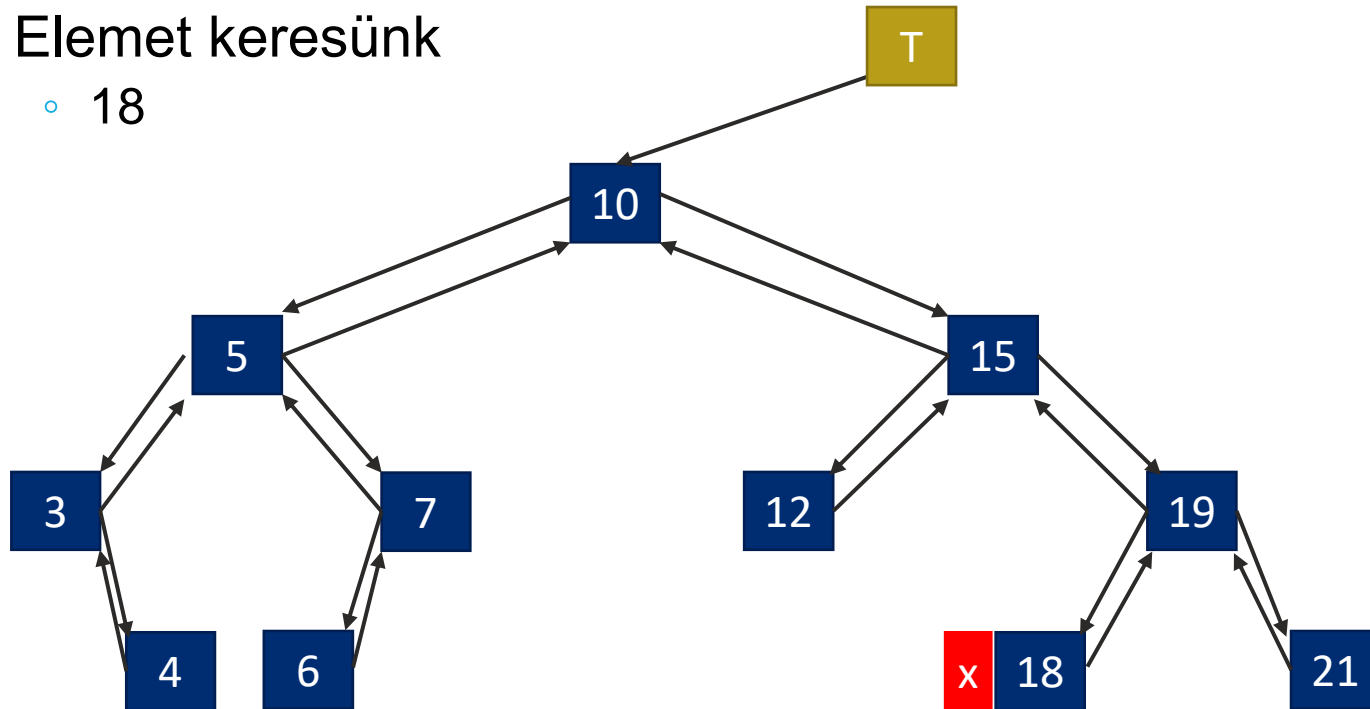


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 18

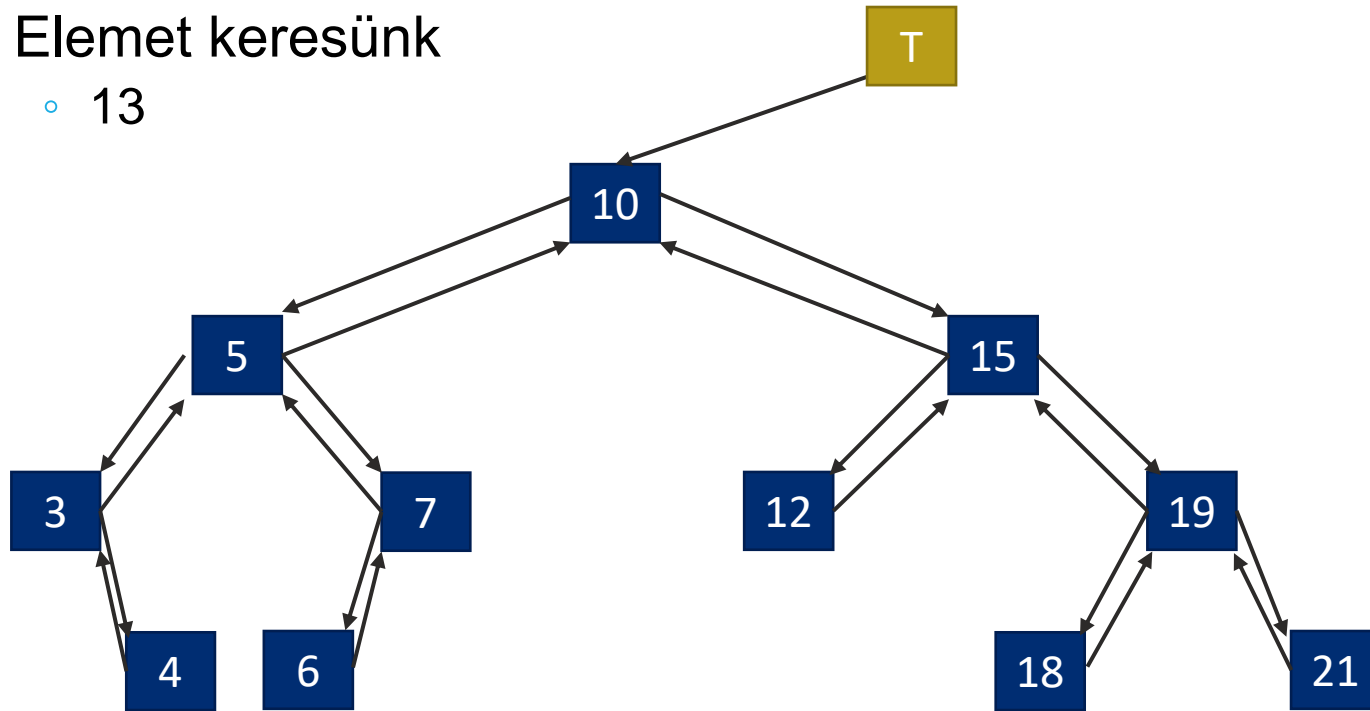


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

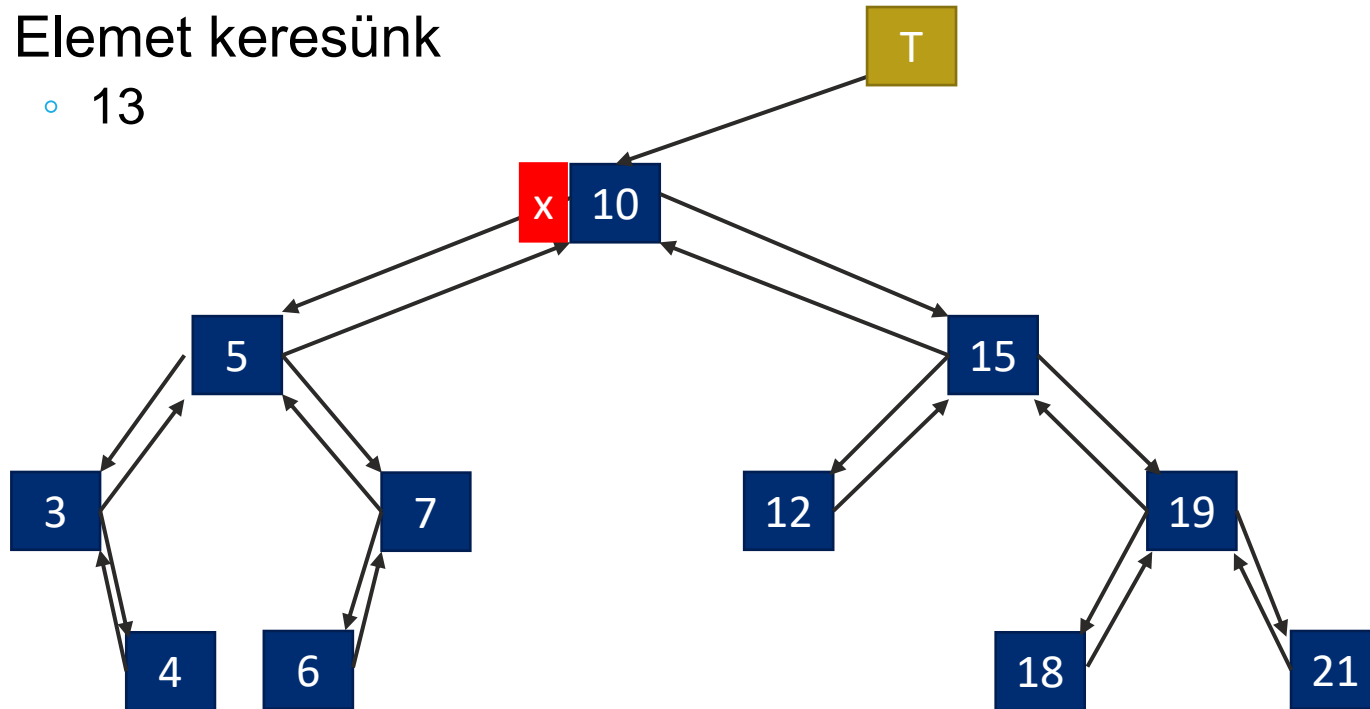


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

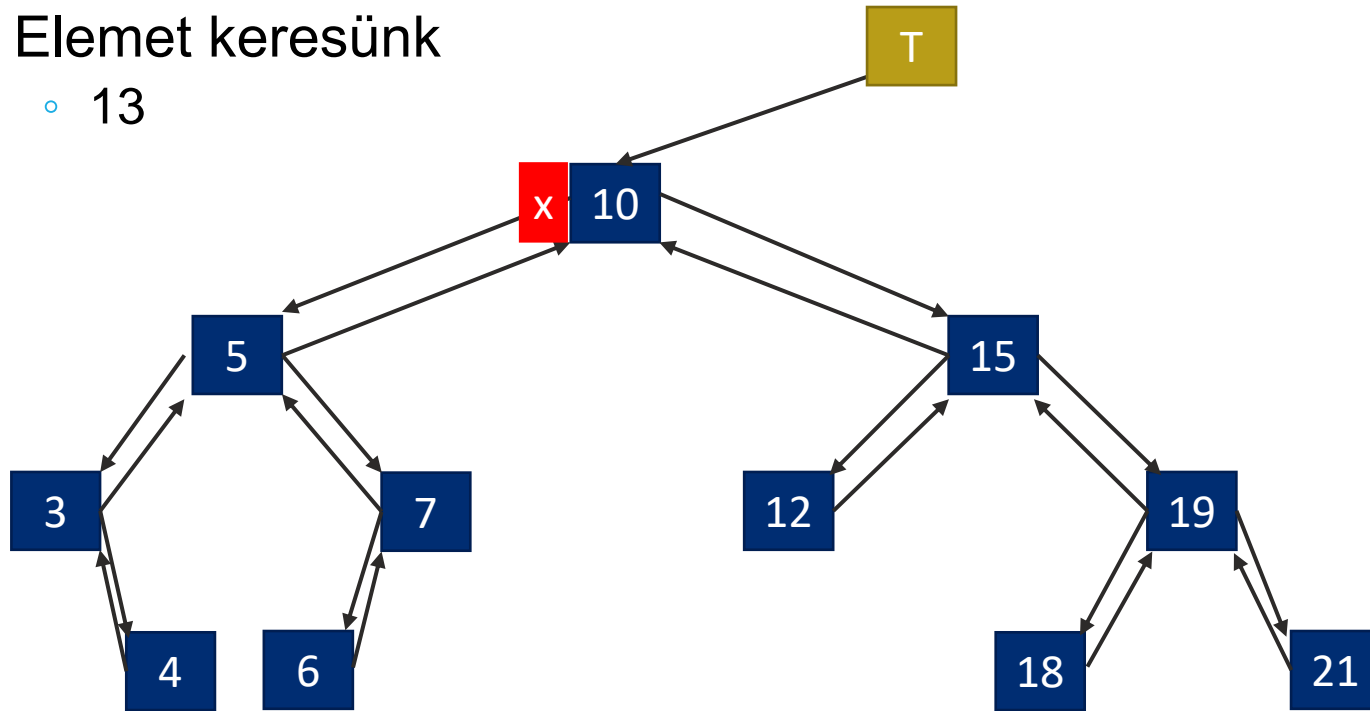


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

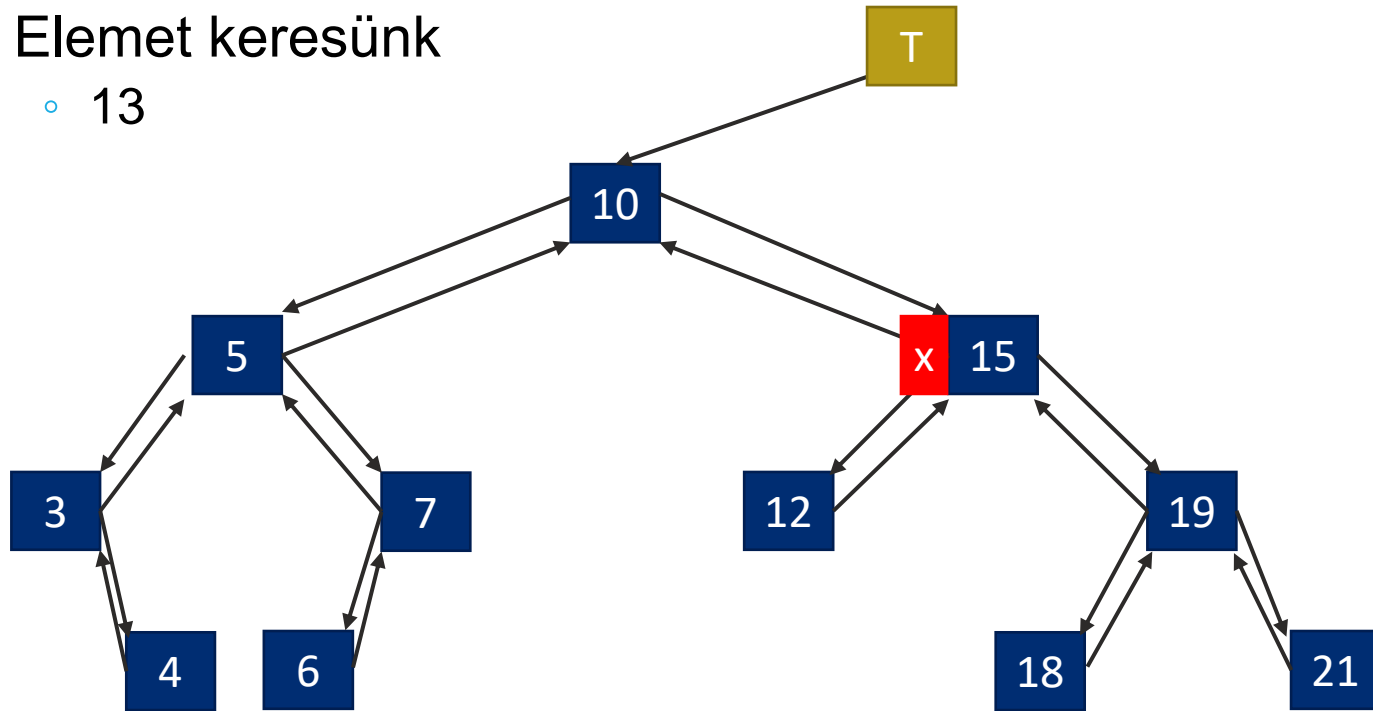


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
        else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13



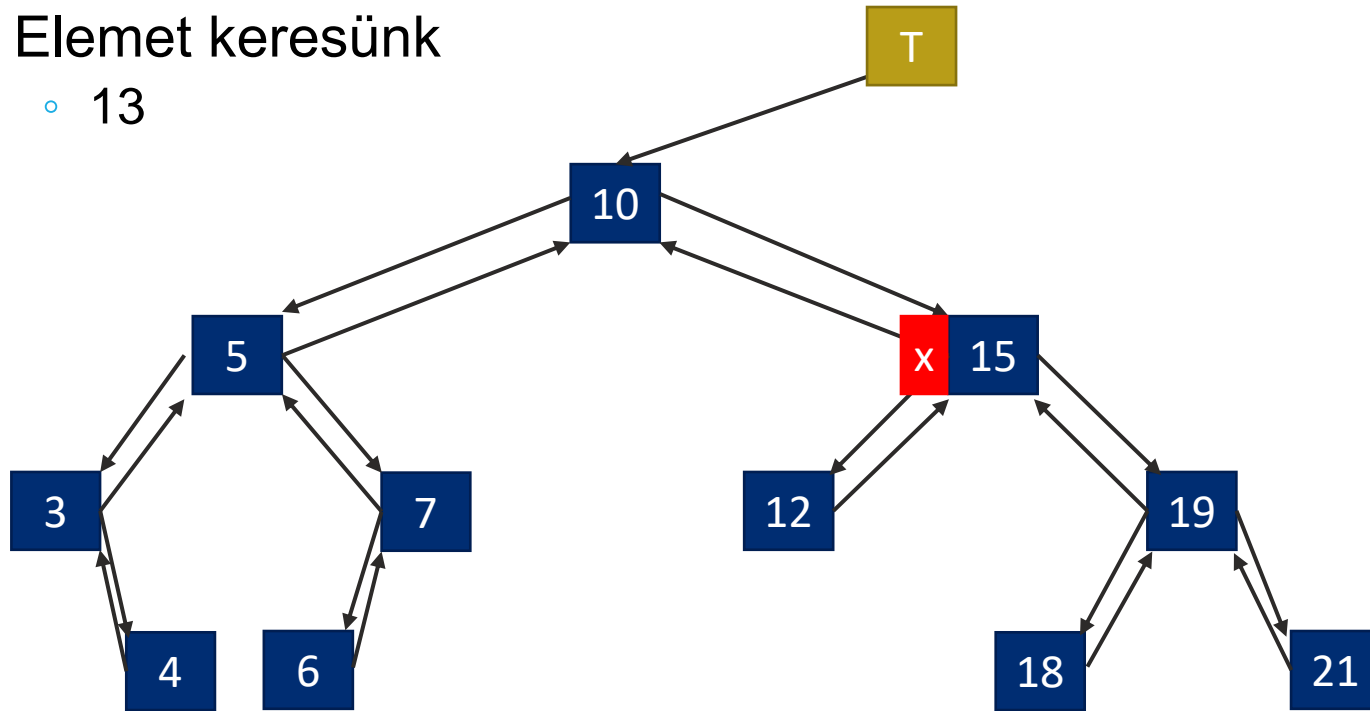
```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```



# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

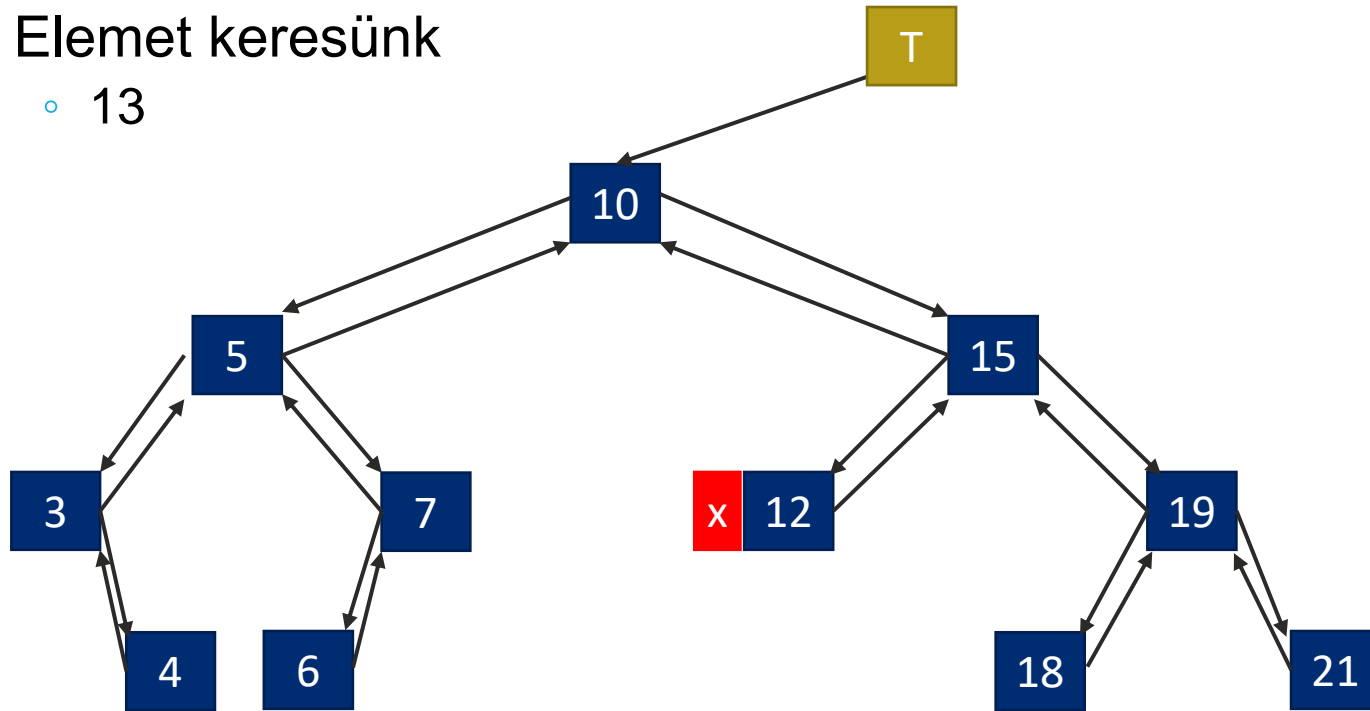


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

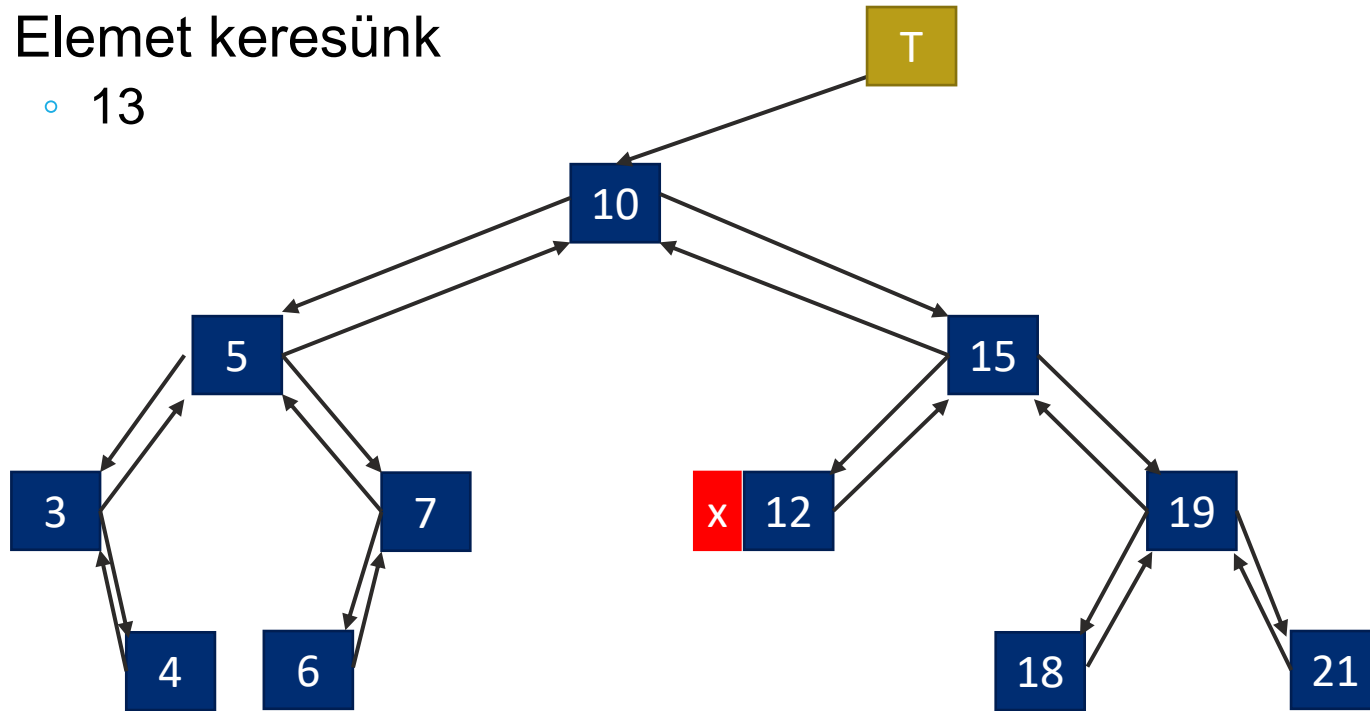


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
    then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

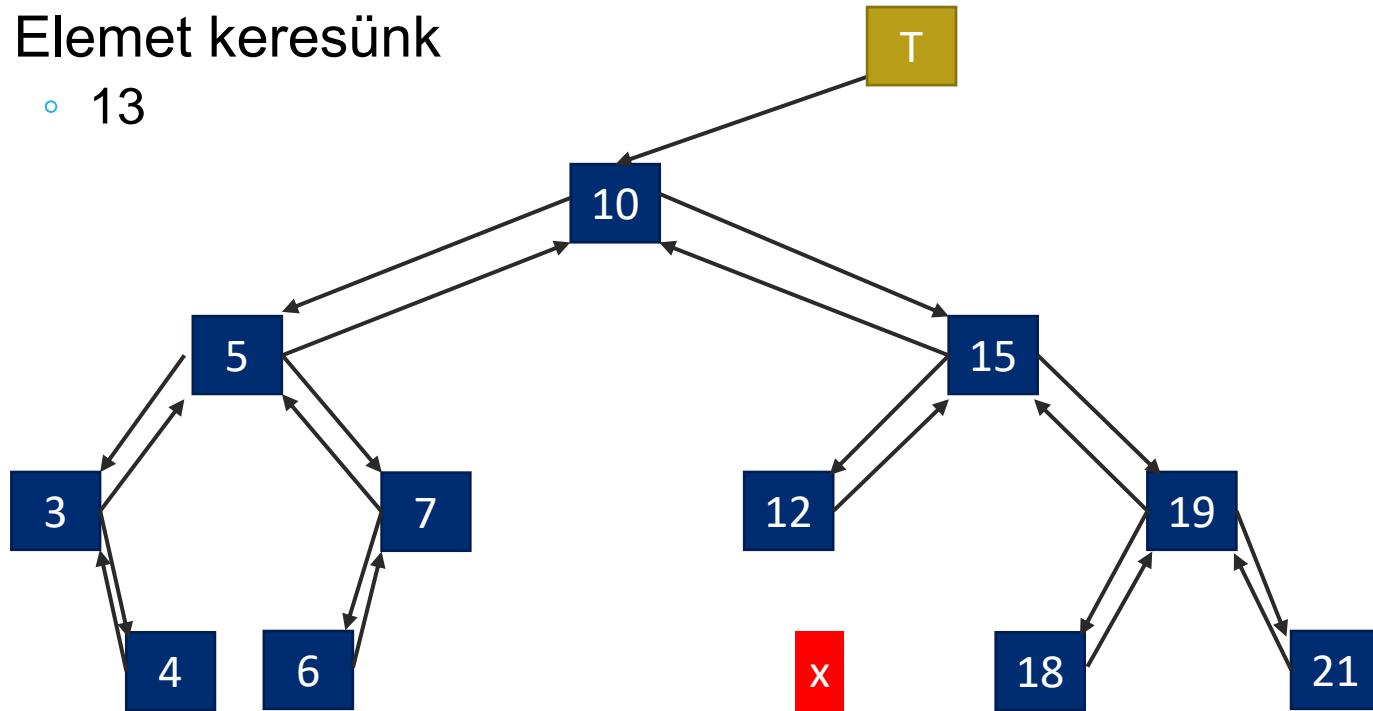
- 13



```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

## Elemet keresünk

- 13

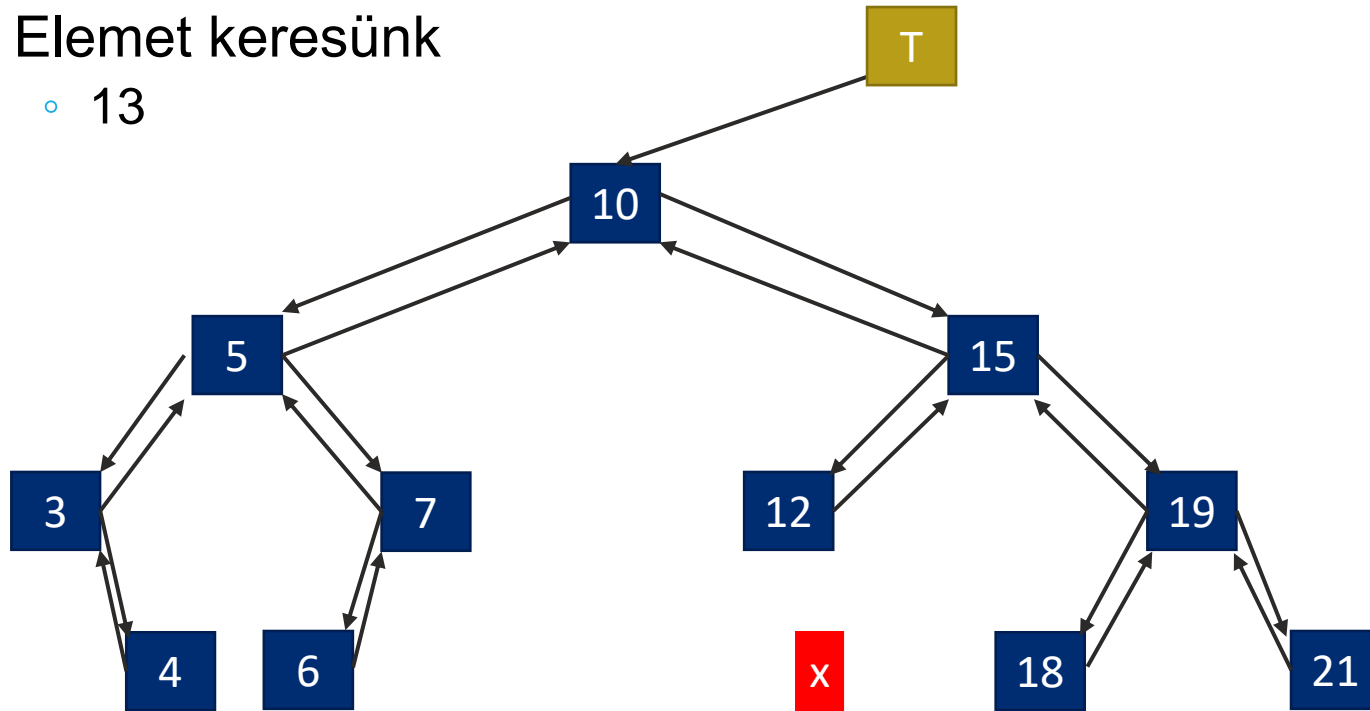


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

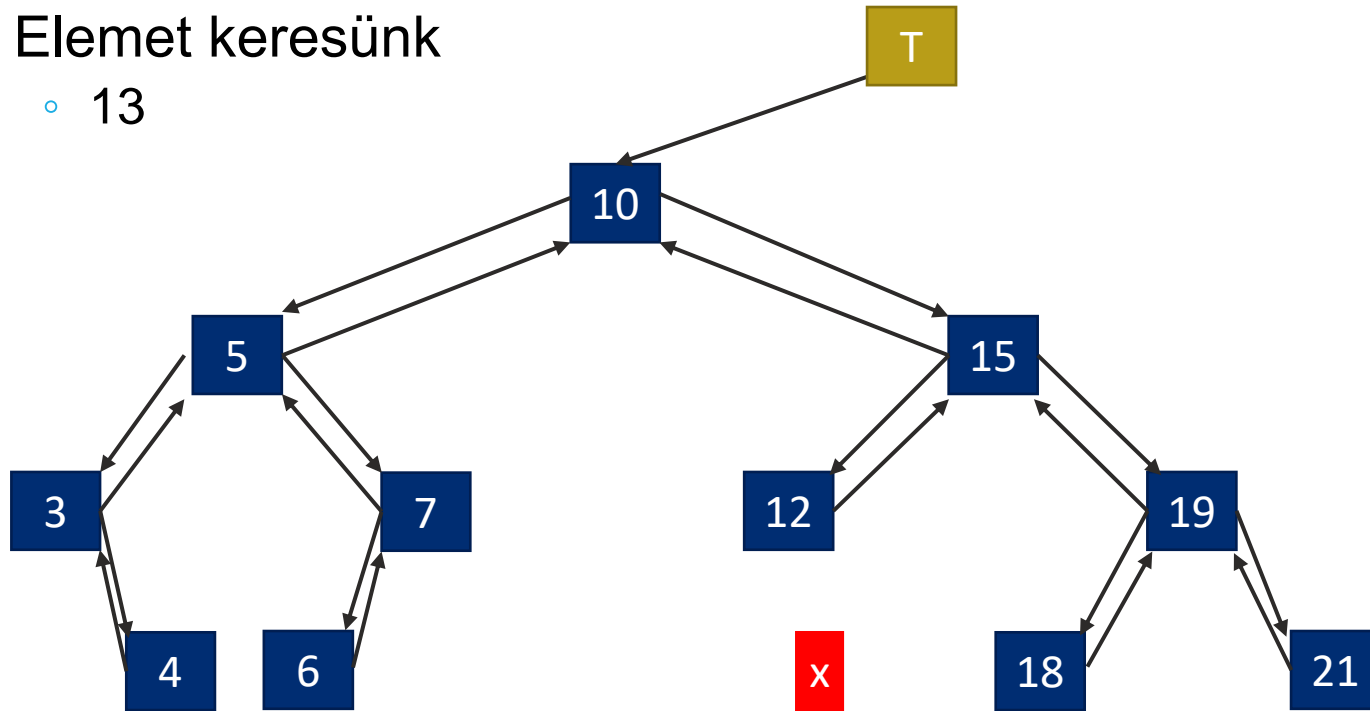


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

# Bináris keresési fa – Elem keresése

Elemet keresünk

- 13

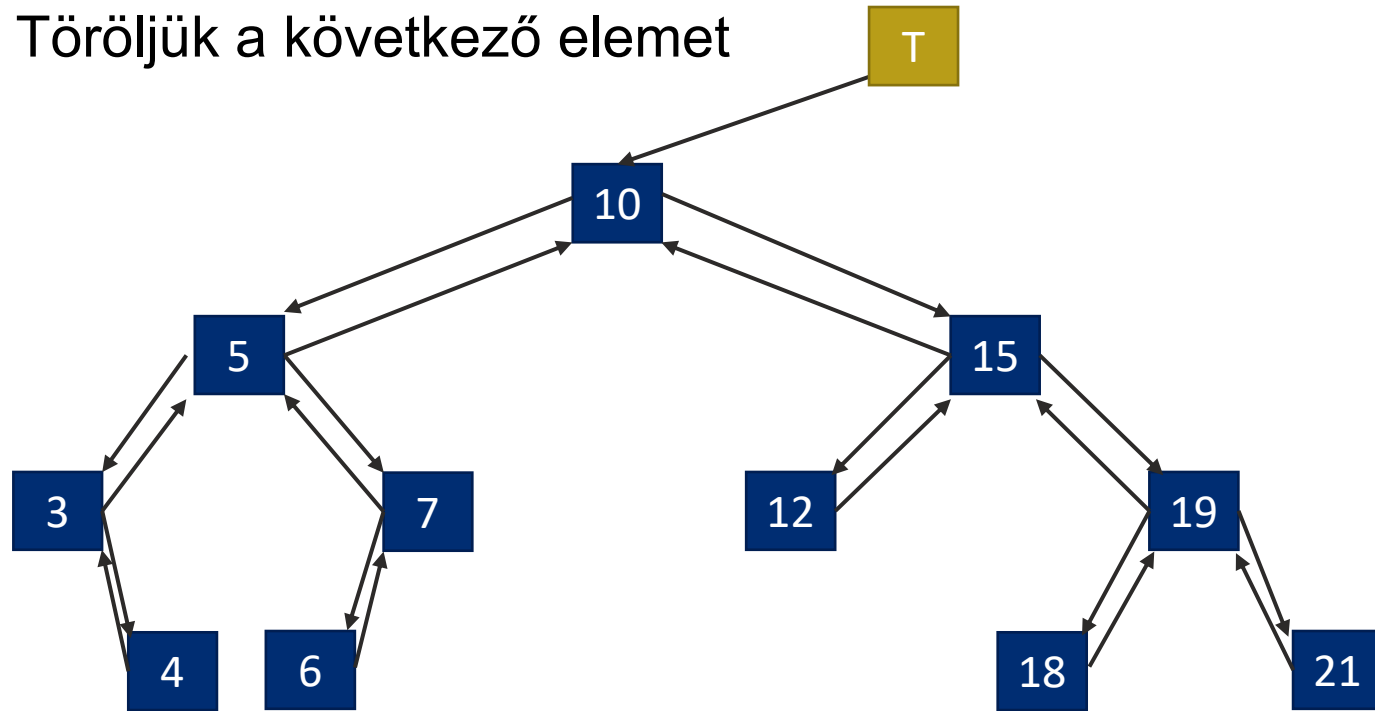


```
Fában-iteratívan-keres(x, k)
while x ≠ NIL and k ≠ kulcs[x] do
    if k < kulcs[x]
        then x ← bal[x]
    else x ← jobb[x]
return x
```

Ilyen értéket nem tárolunk a fában.

# Bináris keresési fa – Elem törlése

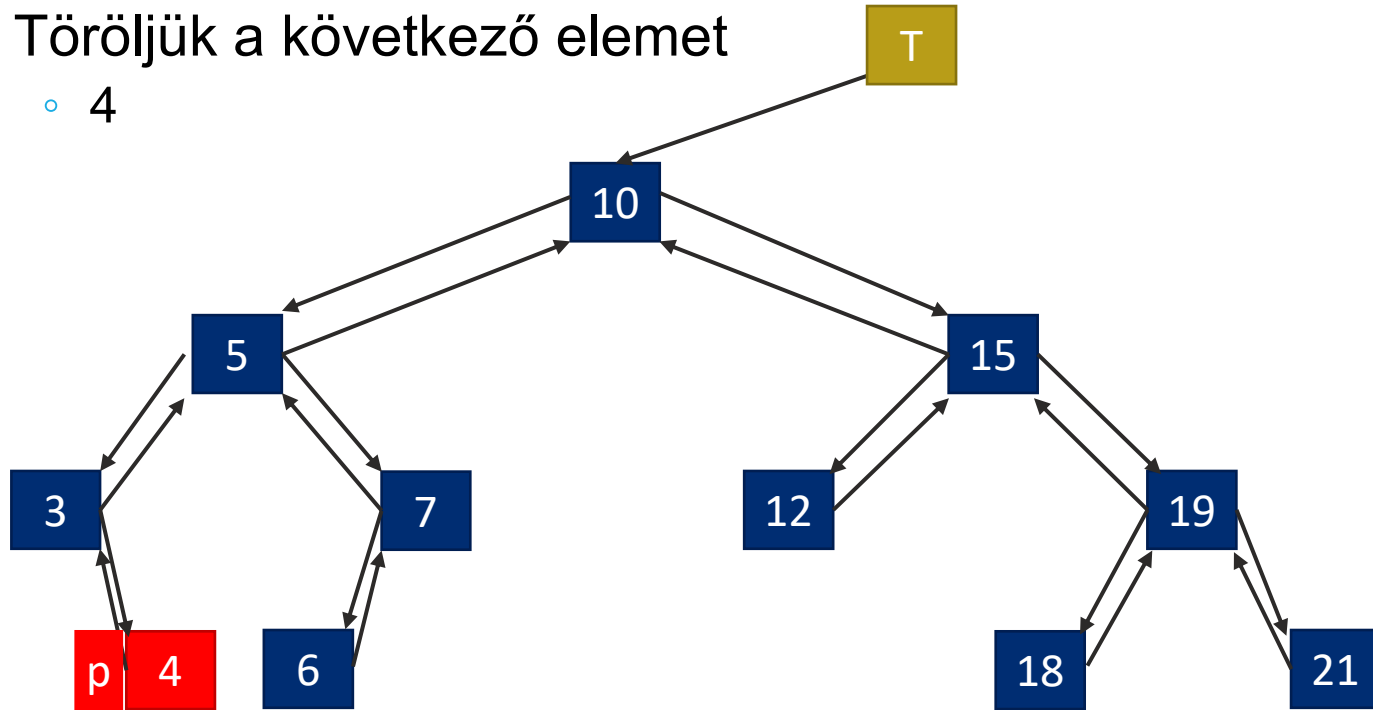
Töröljük a következő elemet



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



Fából-töröl (T,p)

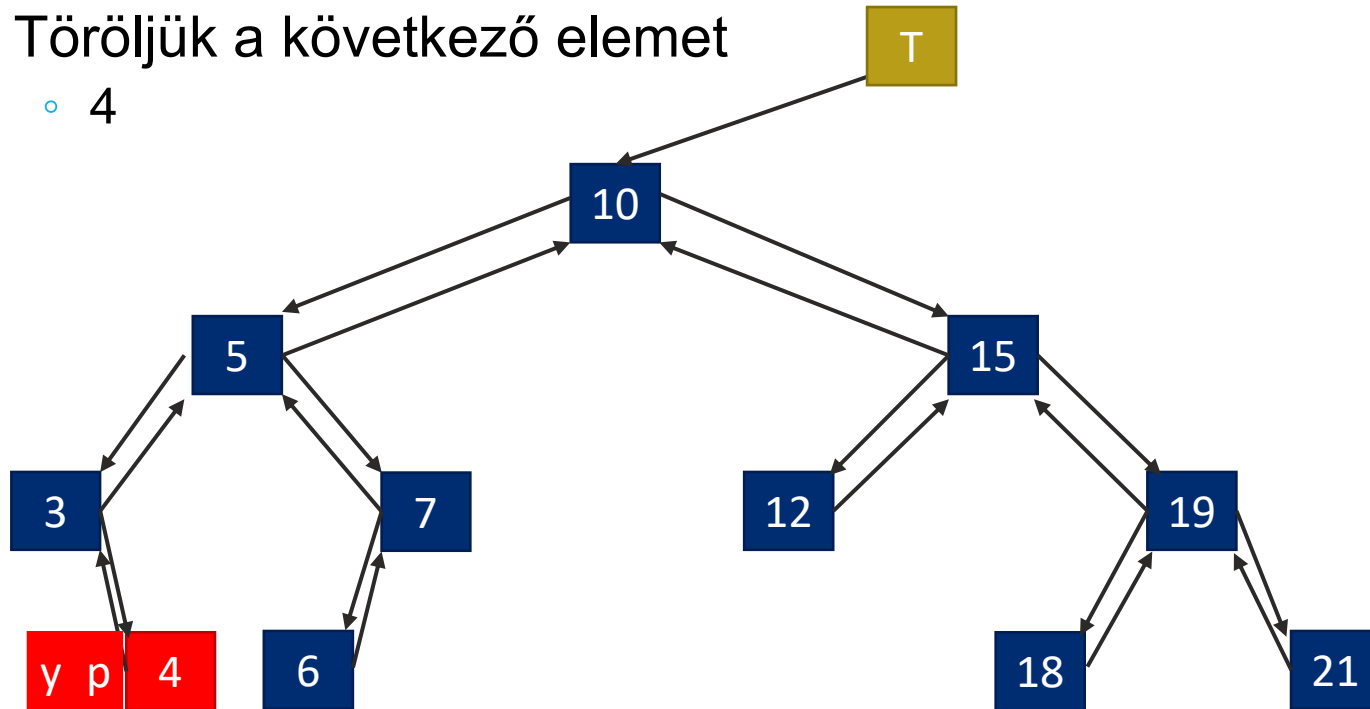
```
if bal[p] = NIL vagy jobb[p] = NIL
  then y ← p
  else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
  then x ← bal[y]
  else x ← jobb[y]
if x ≠ NIL
  then szülő[x] ← szülő[y]
if szülő[y] = NIL
  then gyökér[T] ← x
  else if y = bal[szülő[y]]
    then bal[szülő[y]] ← x
    else jobb[szülő[y]] ← x
if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



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



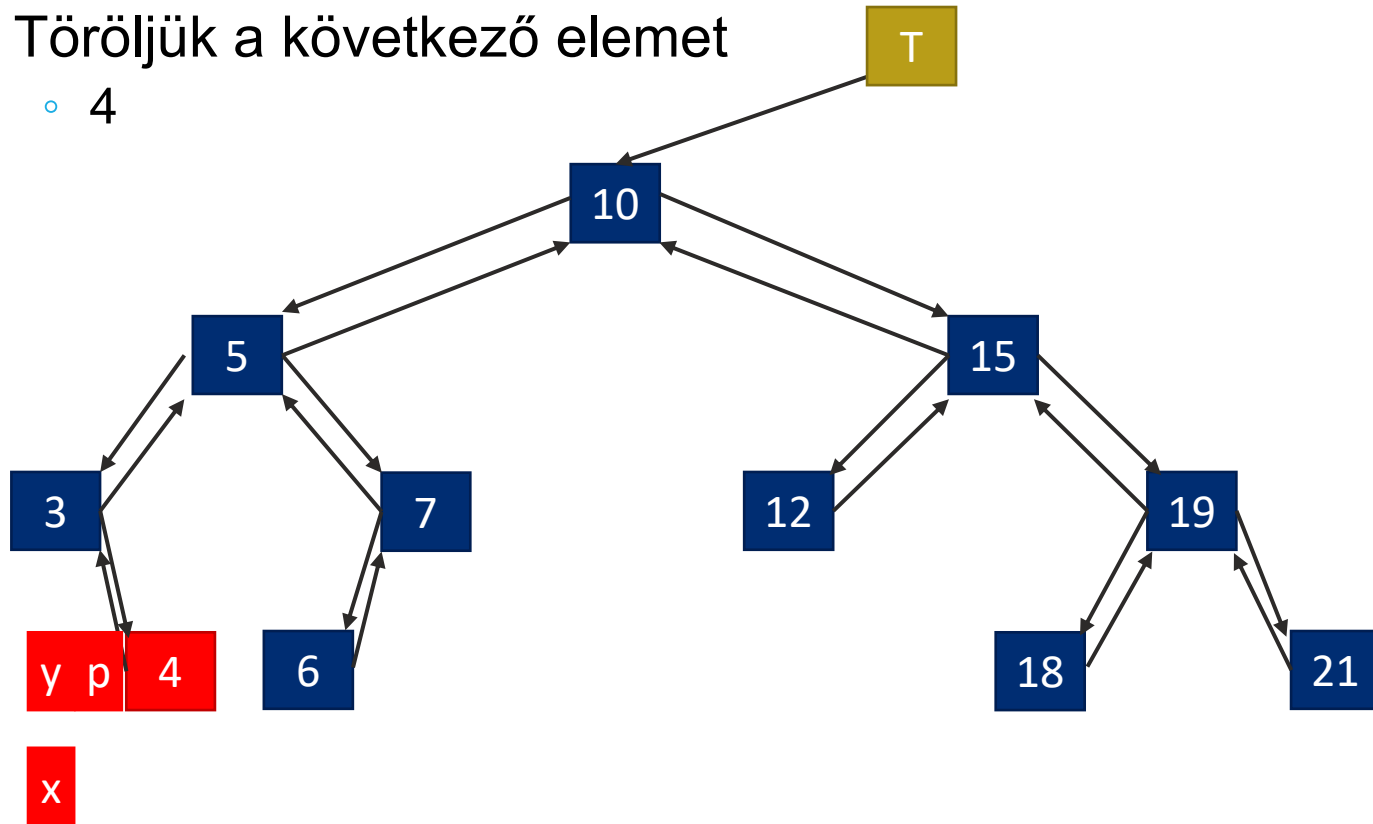
Fából-töröl (T,p)

```
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]
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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



Fából-töröl (T,p)

if bal[p] = NIL vagy jobb[p] = NIL

then  $y \leftarrow p$

else  $y \leftarrow \text{Fában-következő}(T, p)$

if bal[y]  $\neq$  NIL

then  $x \leftarrow \text{bal}[y]$

else  $x \leftarrow \text{jobb}[y]$

if  $x \neq \text{NIL}$

then  $\text{szülő}[x] \leftarrow \text{szülő}[y]$

if  $\text{szülő}[y] = \text{NIL}$

then  $\text{gyökér}[T] \leftarrow x$

else if  $y = \text{bal}[\text{szülő}[y]]$

then  $\text{bal}[\text{szülő}[y]] \leftarrow x$

else  $\text{jobb}[\text{szülő}[y]] \leftarrow x$

if  $y \neq p$

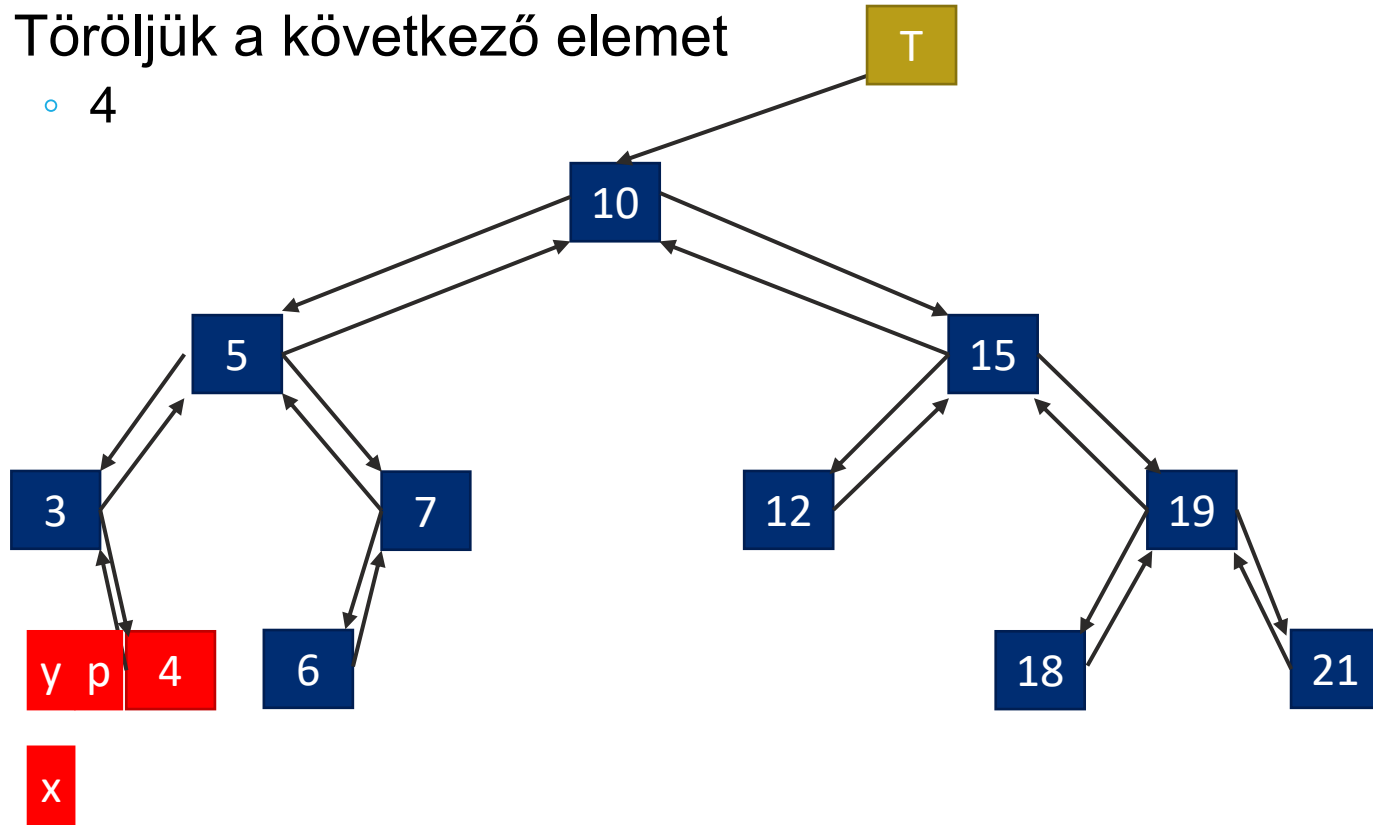
then  $\text{kulcs}[p] \leftarrow \text{kulcs}[y]$

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



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]
```

```
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
```

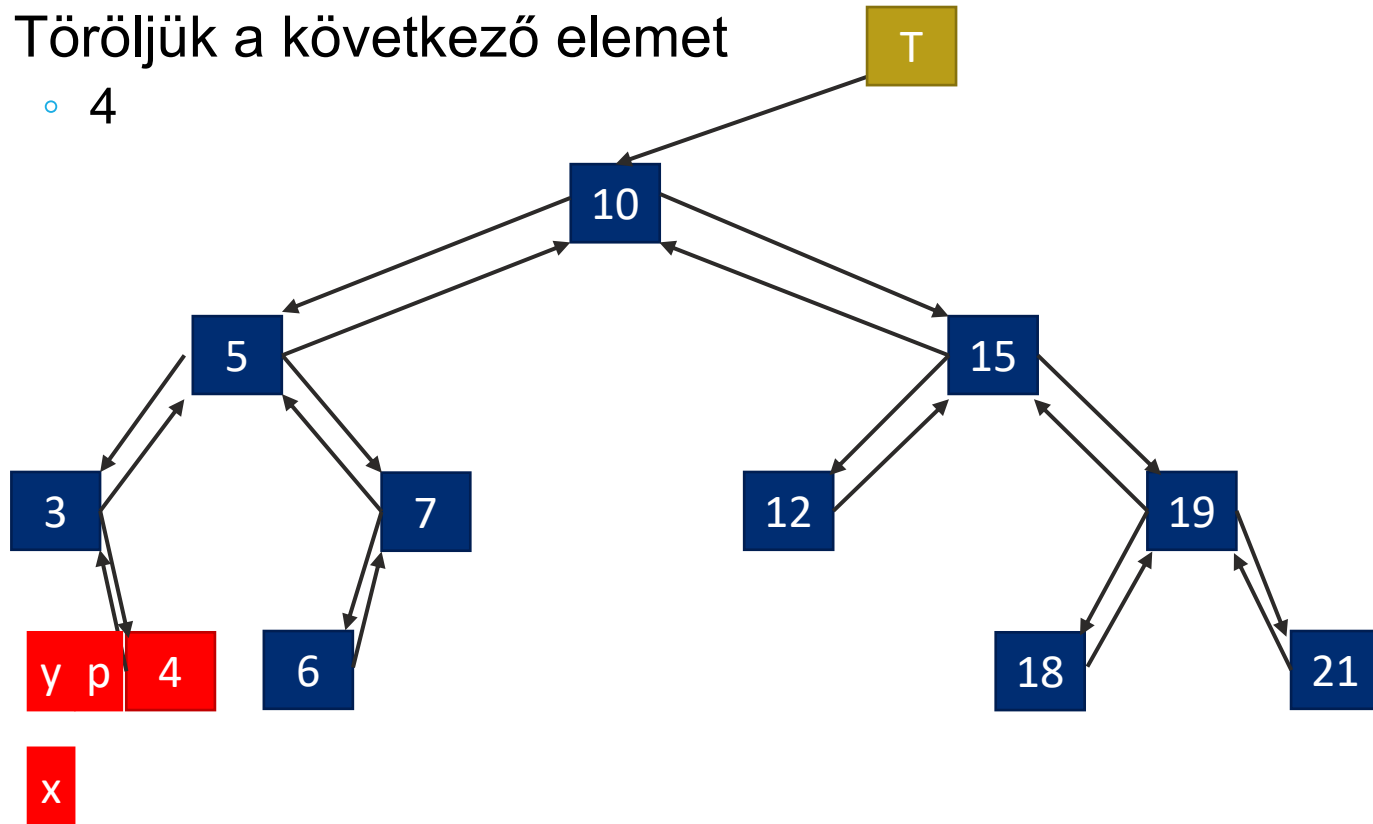
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



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]
```

```
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
```

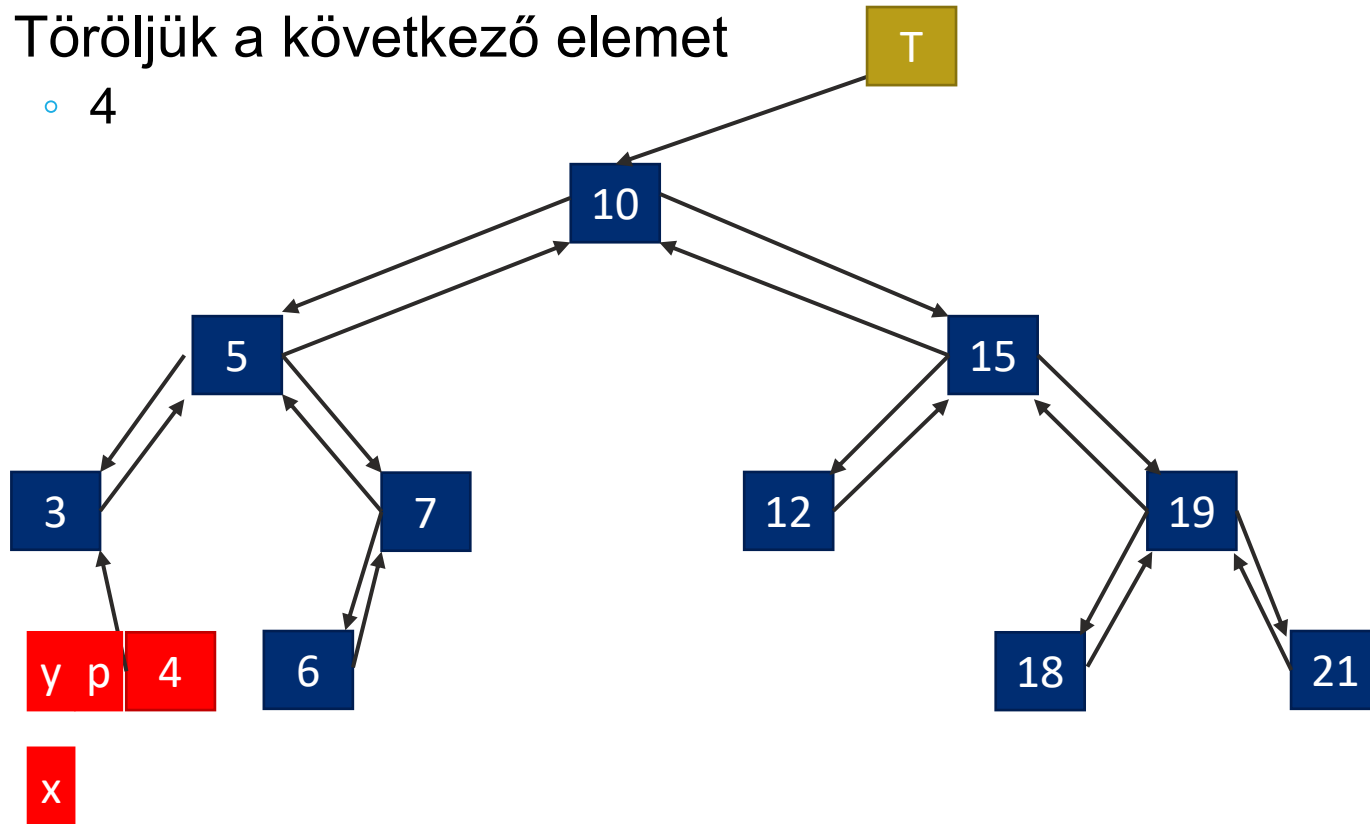
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



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]

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

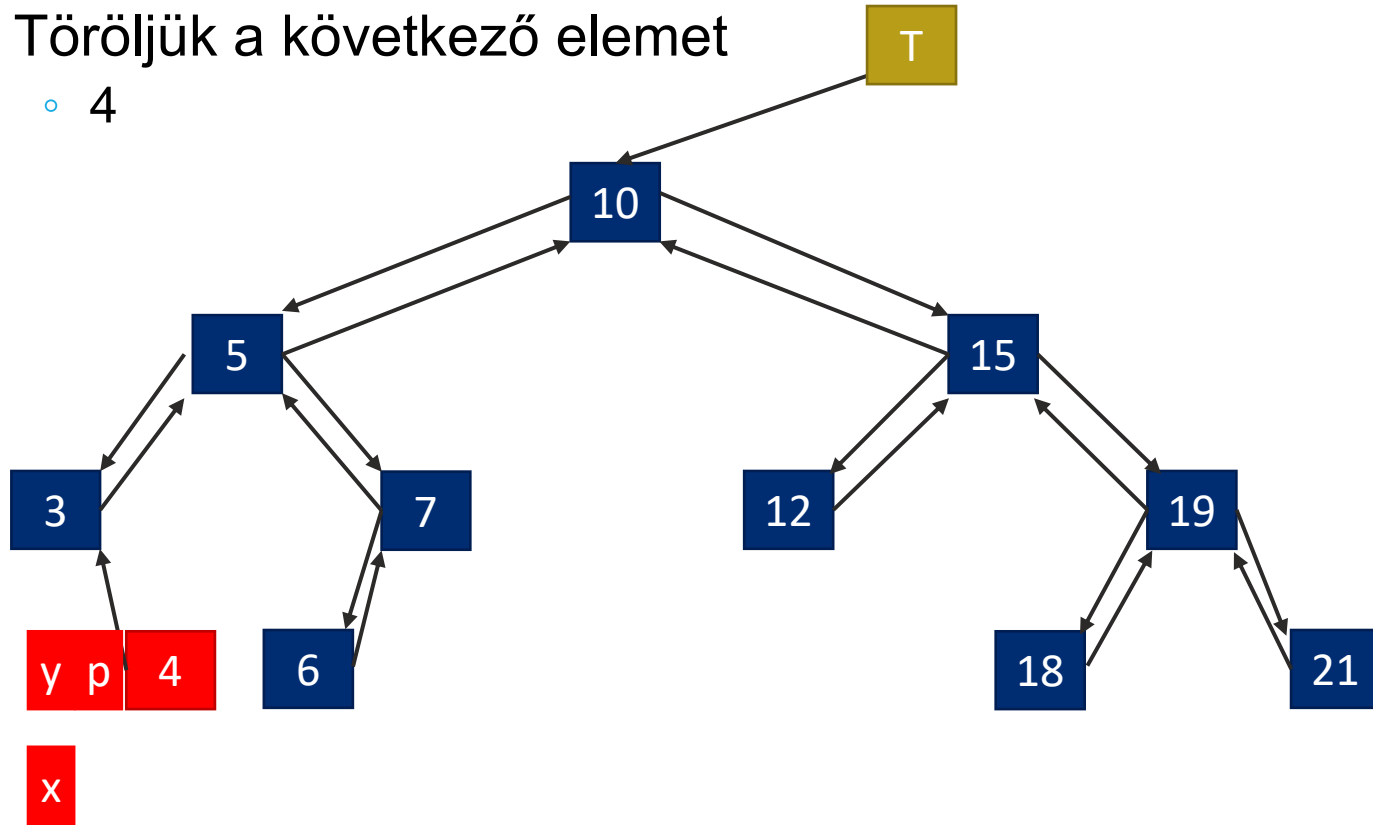
then kulcs[p] ← kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



Fából-töröl (T,p)

if bal[p] = NIL vagy jobb[p] = NIL

then  $y \leftarrow p$

else  $y \leftarrow$  Fában-következő(T, p)

if bal[y]  $\neq$  NIL

then  $x \leftarrow$  bal[y]

else  $x \leftarrow$  jobb[y]

if  $x \neq$  NIL

then  $szülő[x] \leftarrow szülő[y]$

if  $szülő[y] =$  NIL

then  $gyökér[T] \leftarrow x$

else if  $y =$  bal[ $szülő[y]$ ]

then bal[ $szülő[y]$ ]  $\leftarrow x$

else jobb[ $szülő[y]$ ]  $\leftarrow x$

if  $y \neq p$

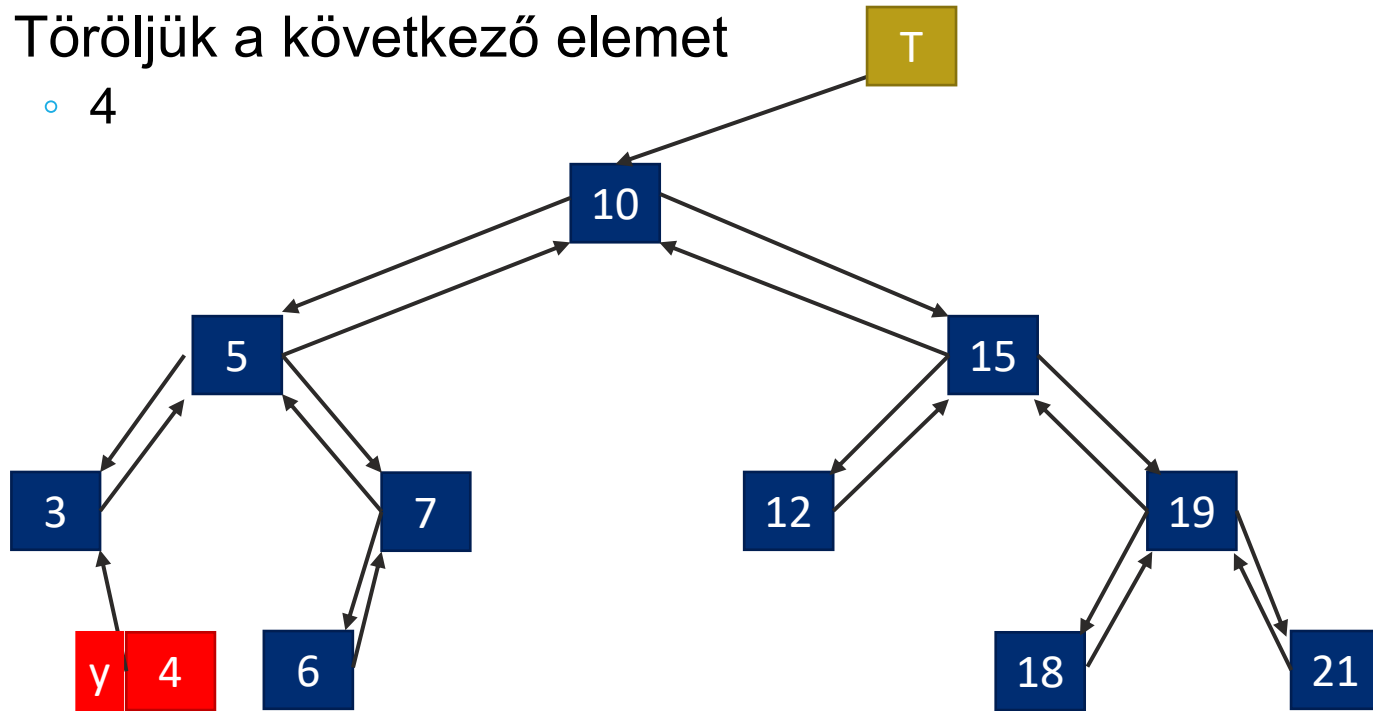
then kulcs[p]  $\leftarrow$  kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 4



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]
```

```
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
```

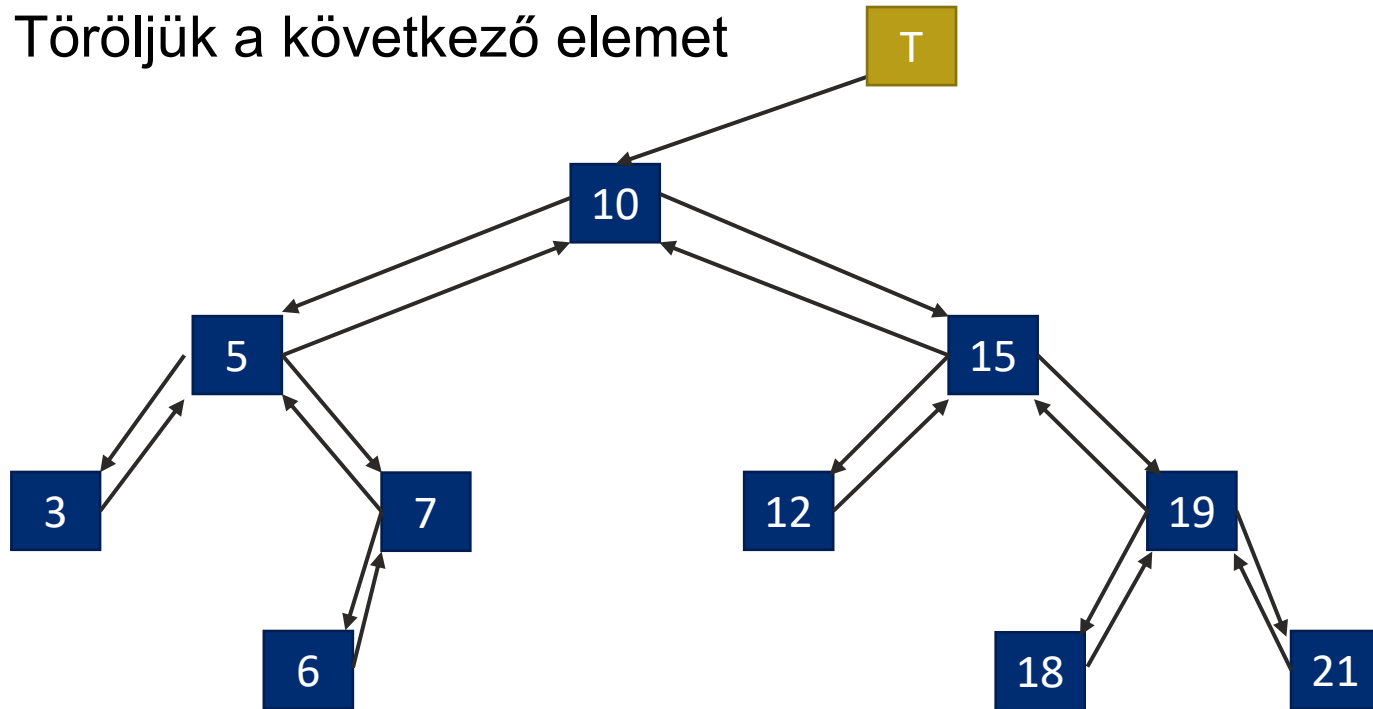
```
if y ≠ p
```

```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet



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]
```

```
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]
```

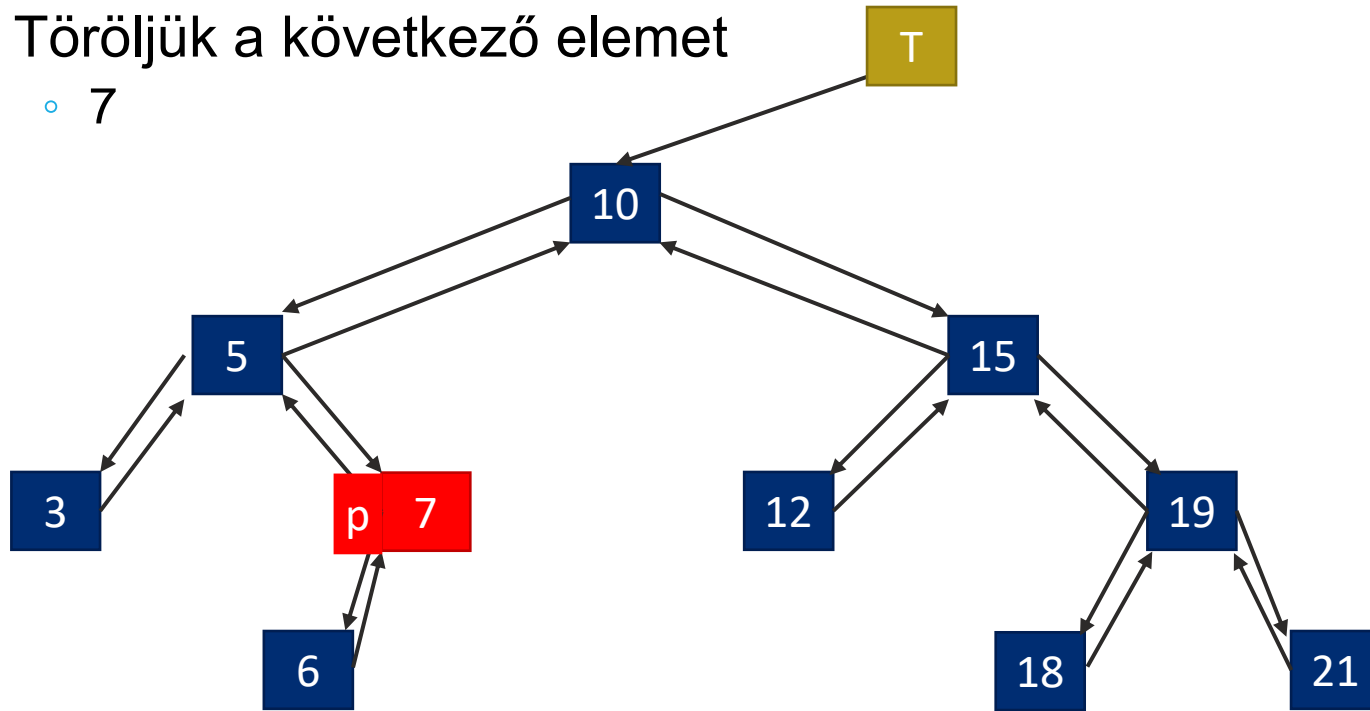
```
return y
```



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



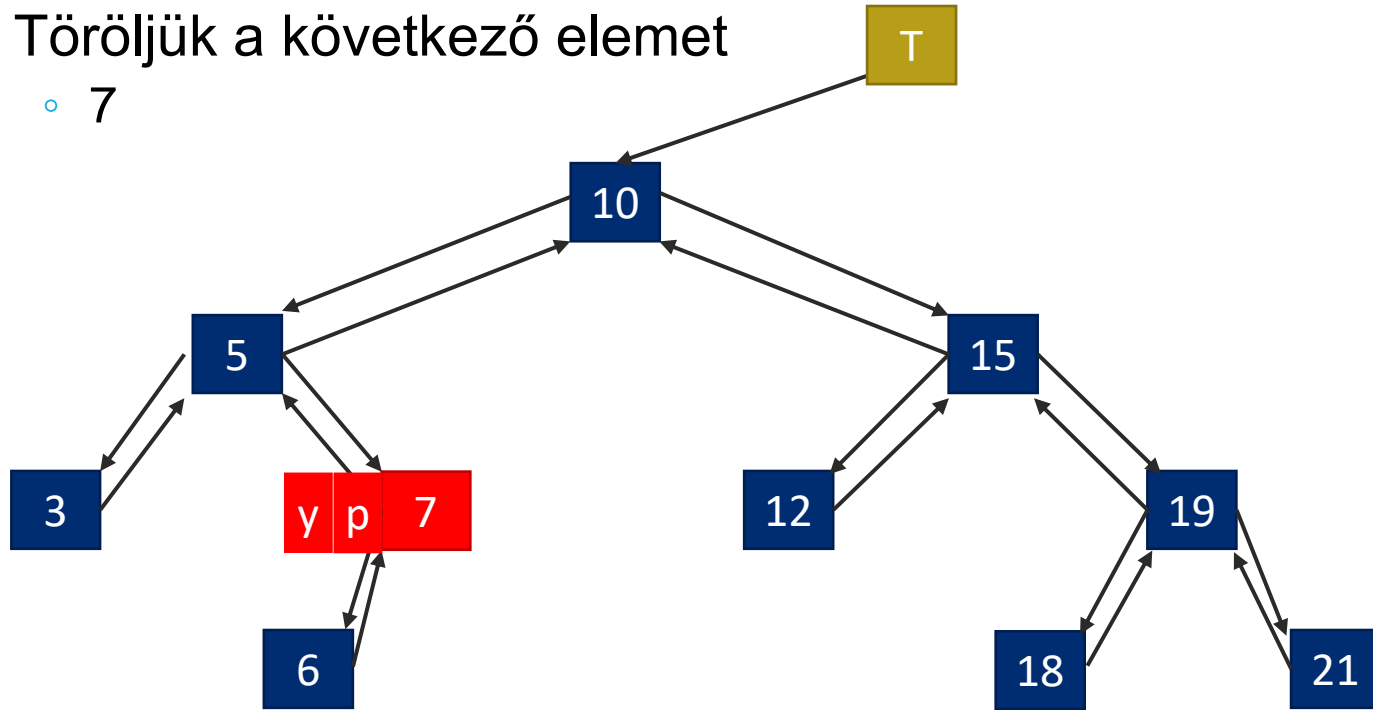
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



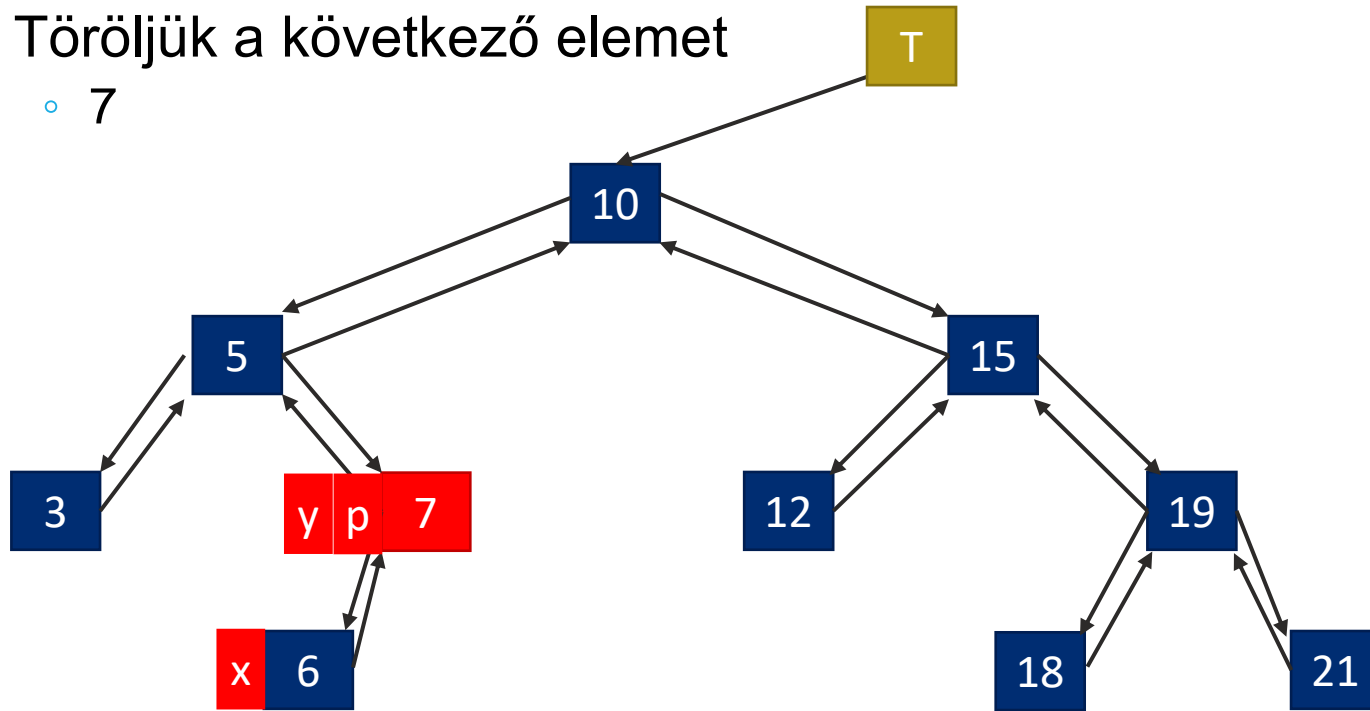
Fából-töröl (T,p)

```
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]
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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



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]

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

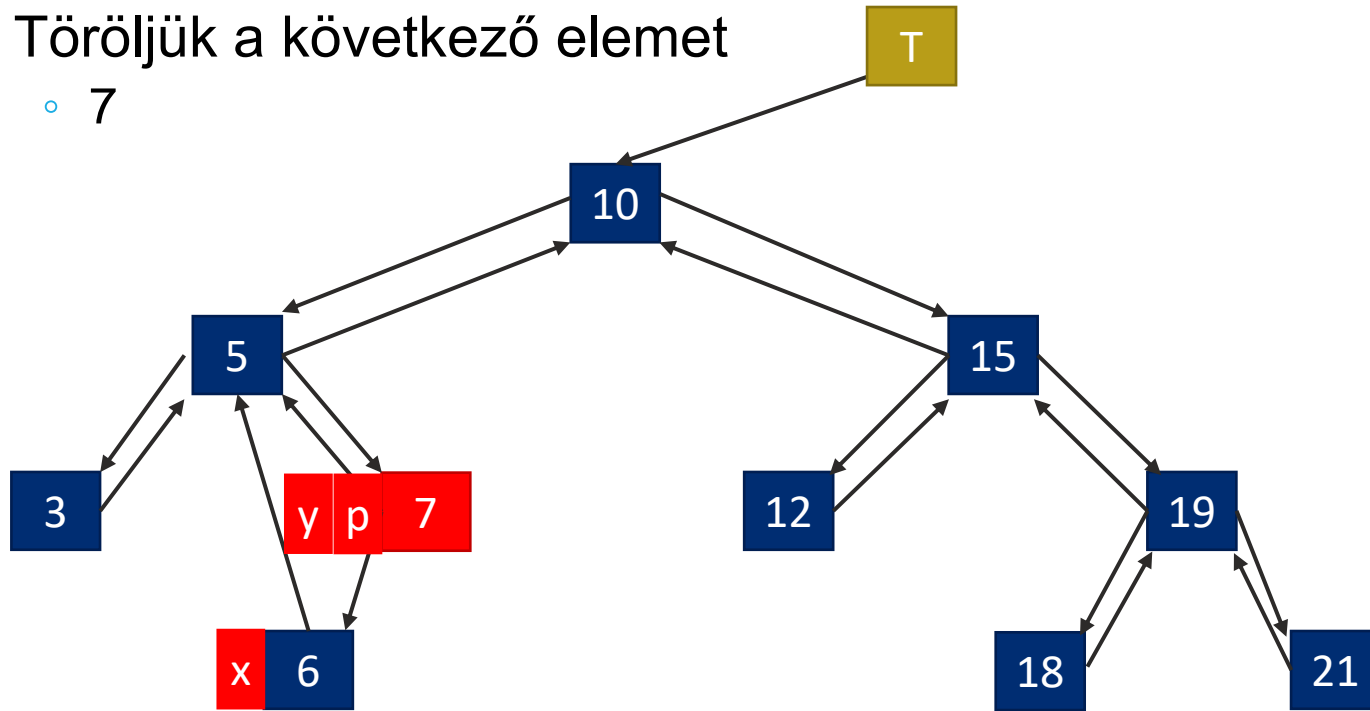
then kulcs[p] ← kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



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]
```

```
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
```

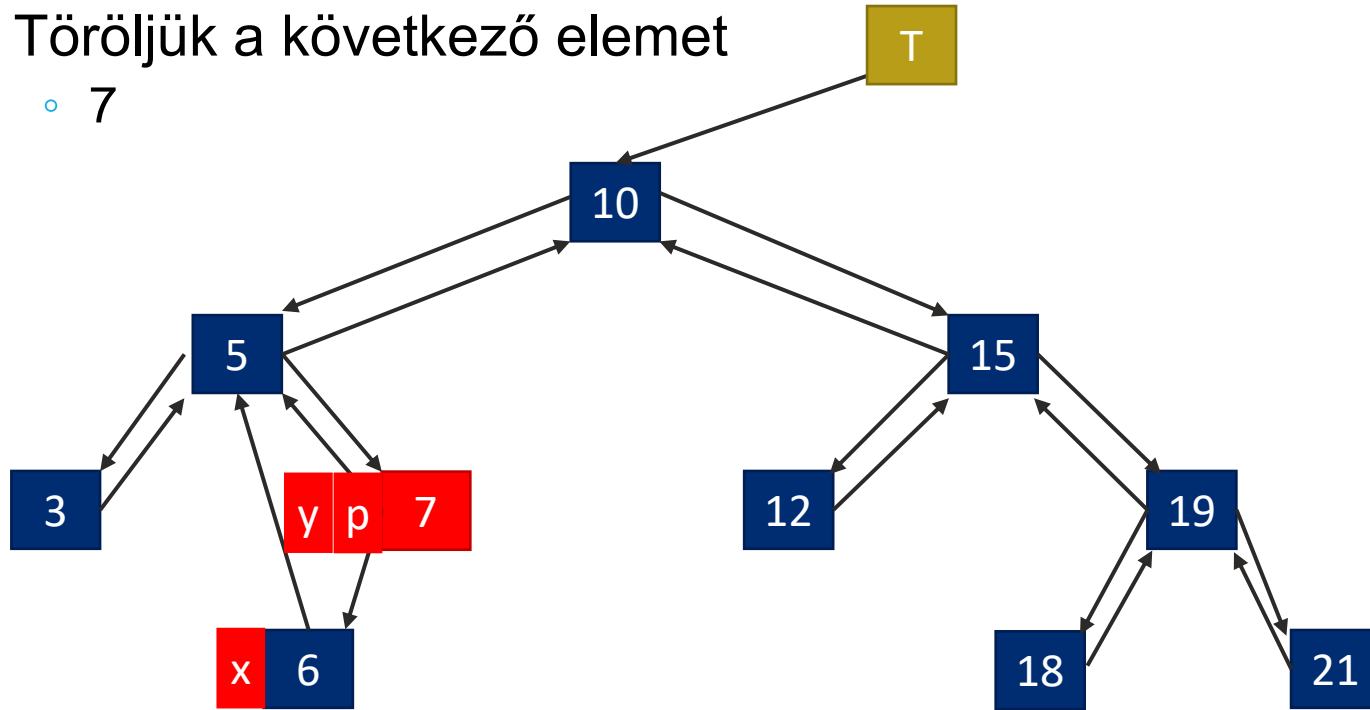
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



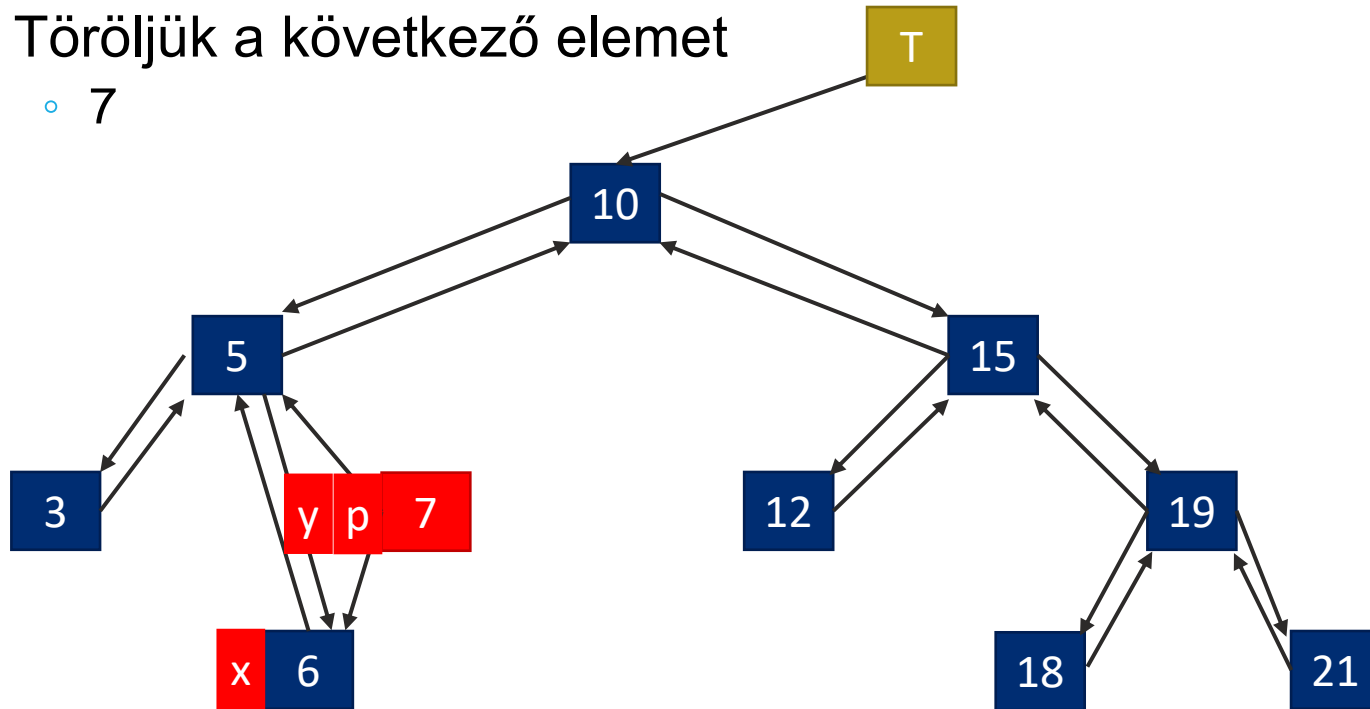
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



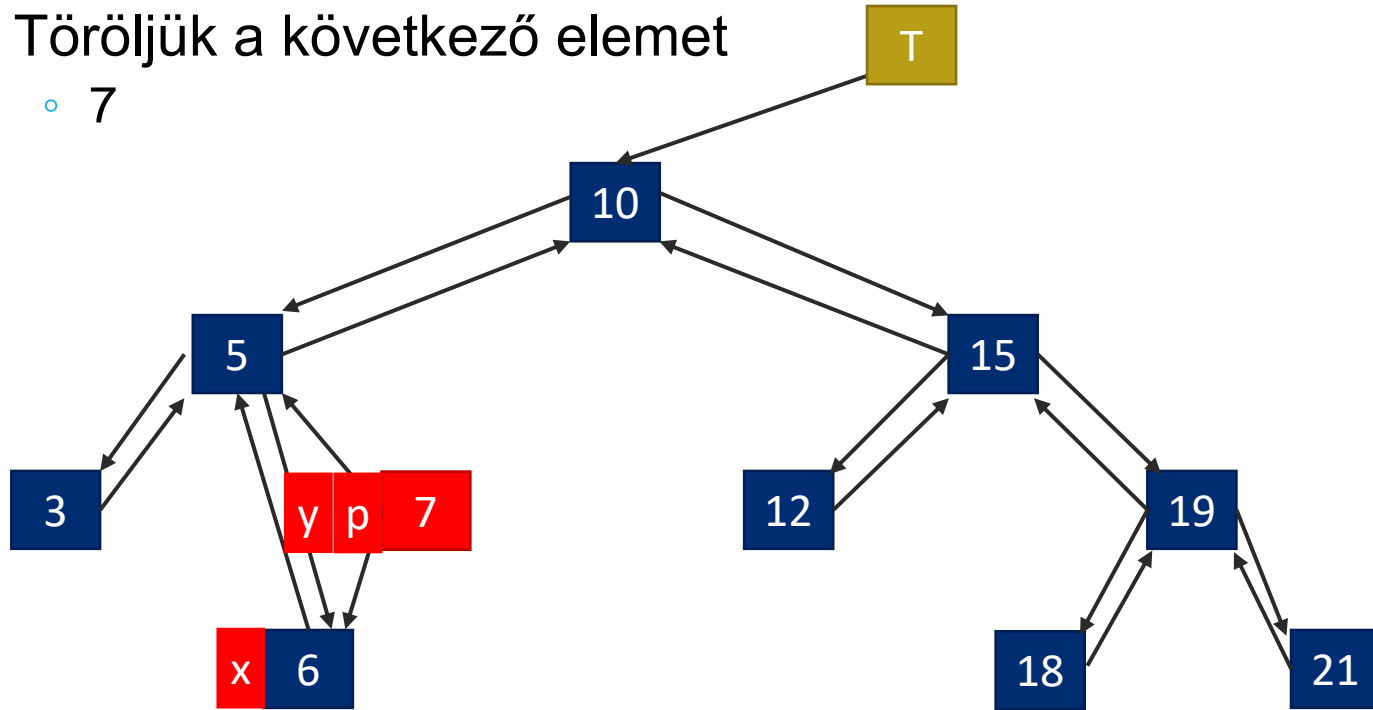
Fából-töröl (T,p)

```
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]
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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



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]
```

```
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
```

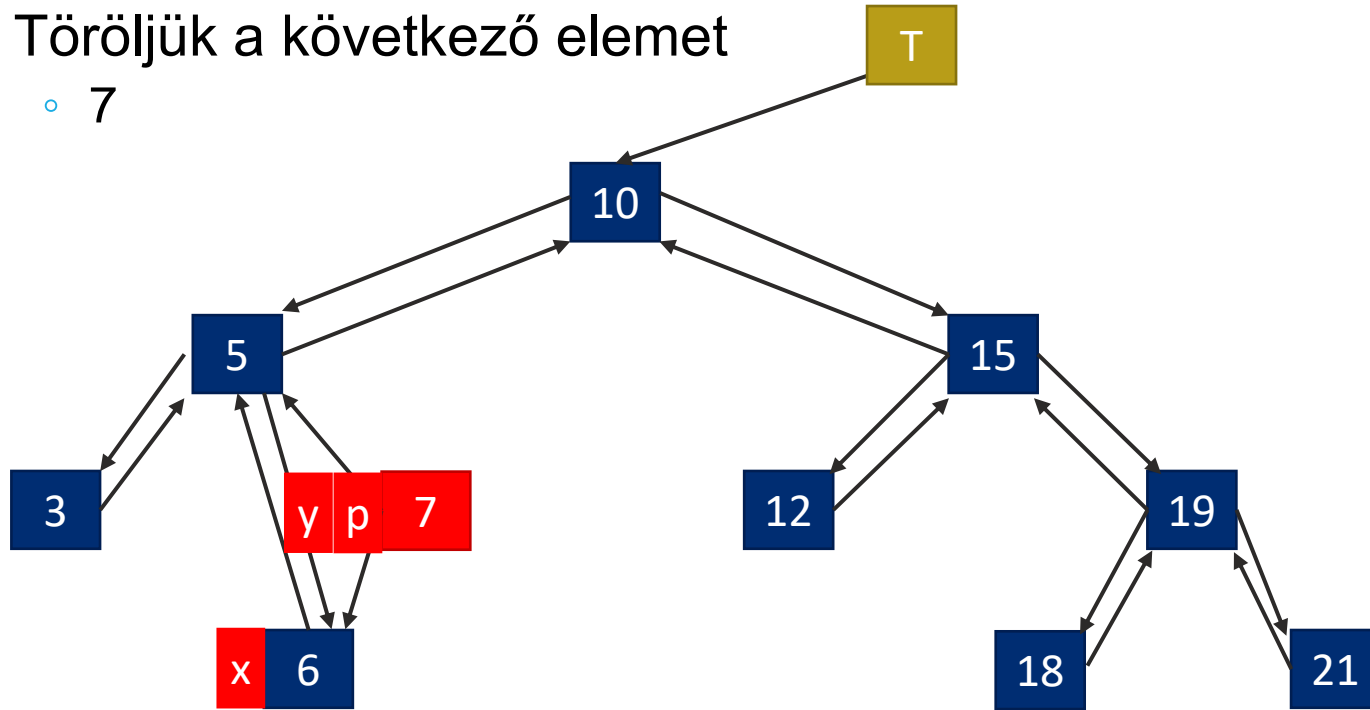
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 7



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]
```

```
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
```

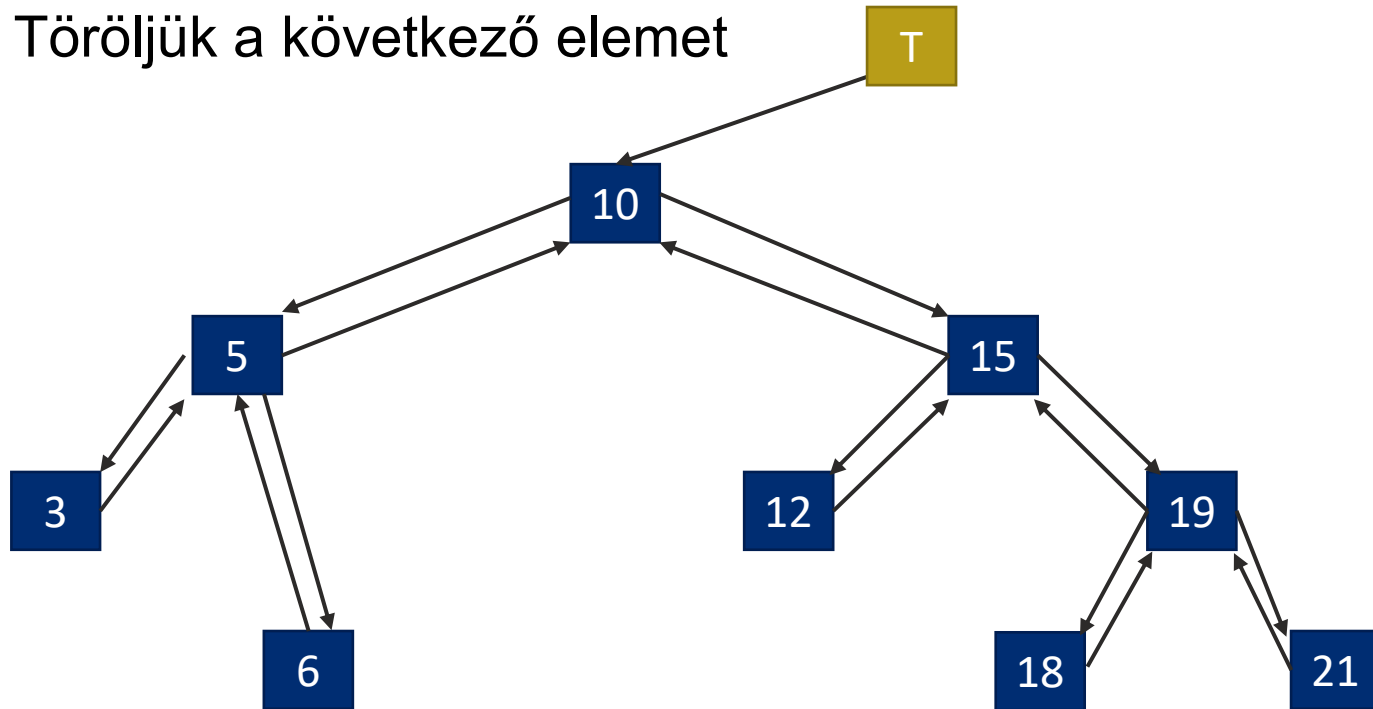
```
then kulcs[p] ← kulcs[y]
```

```
return y
```



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

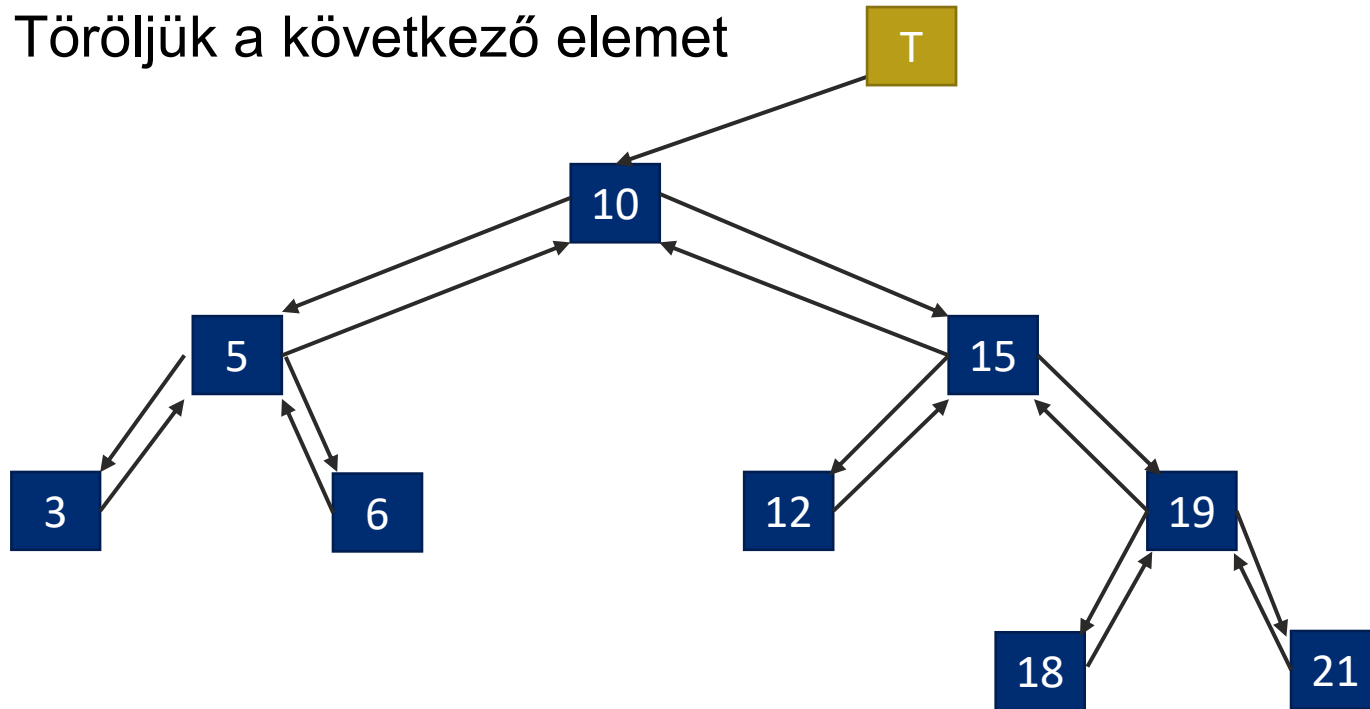


Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet



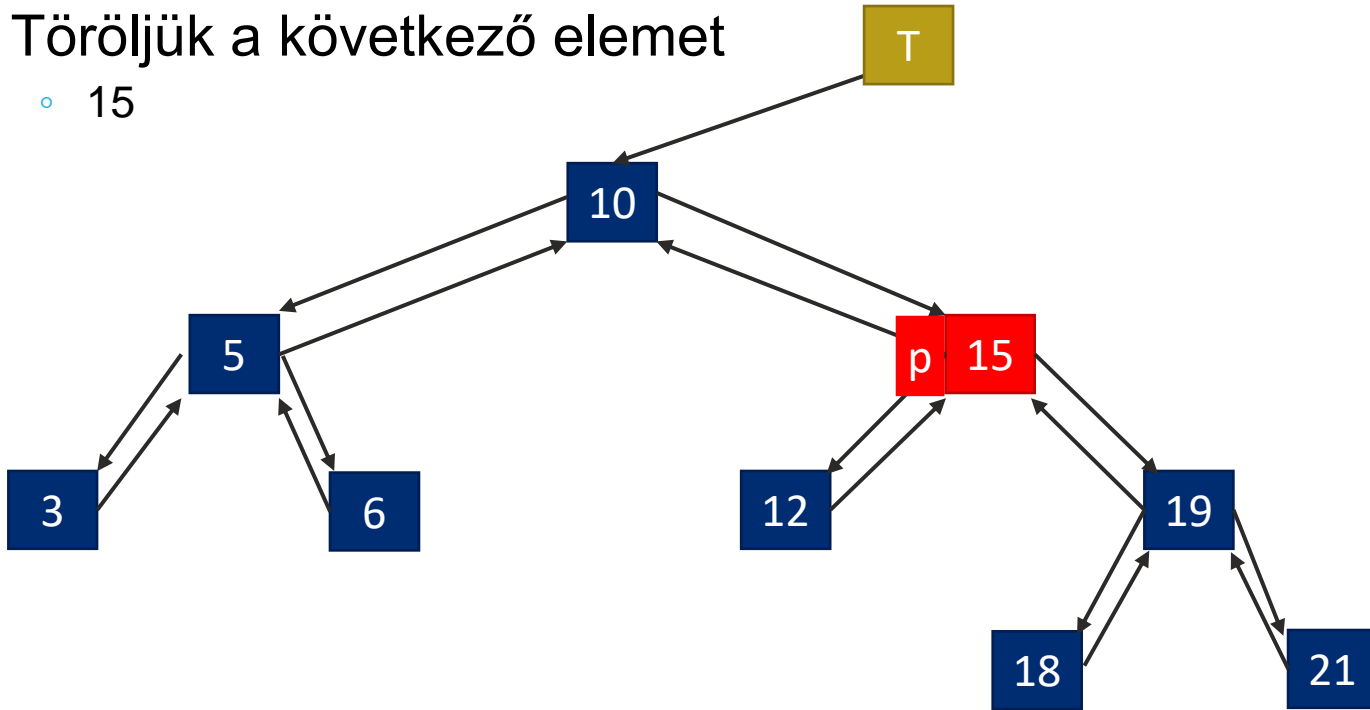
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



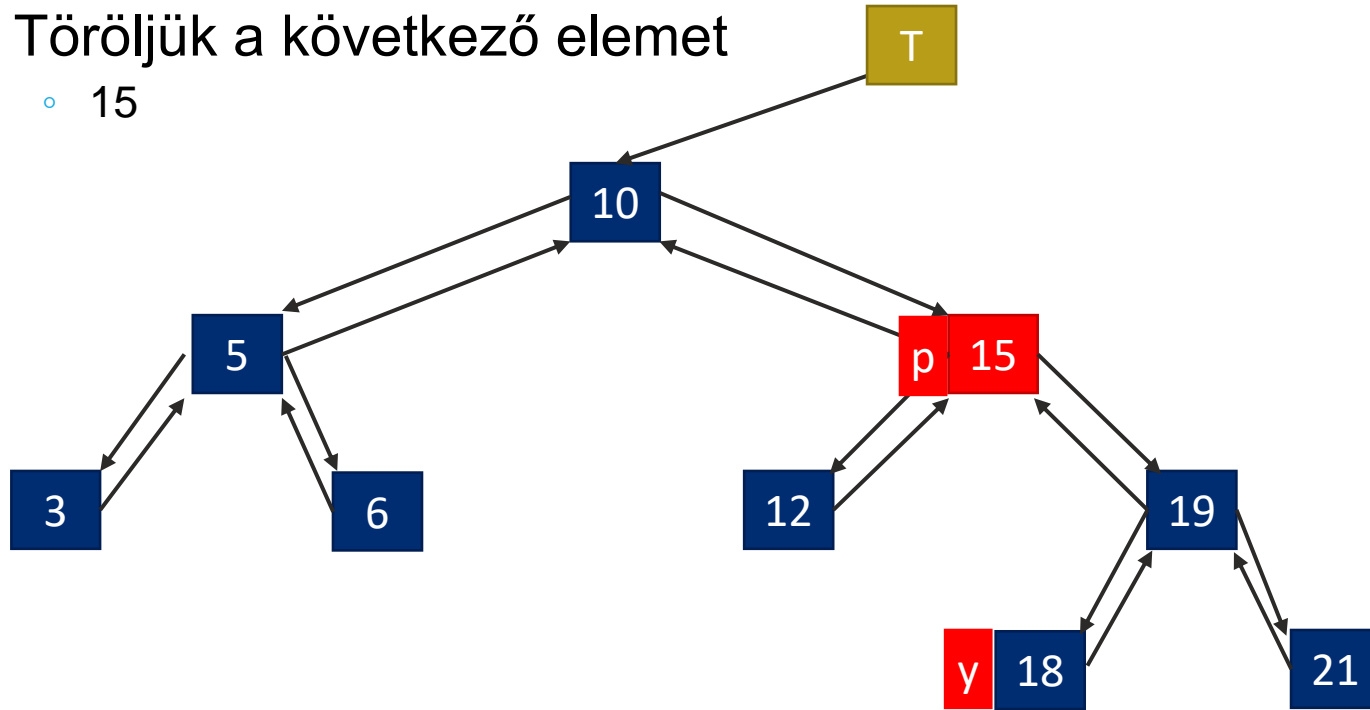
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



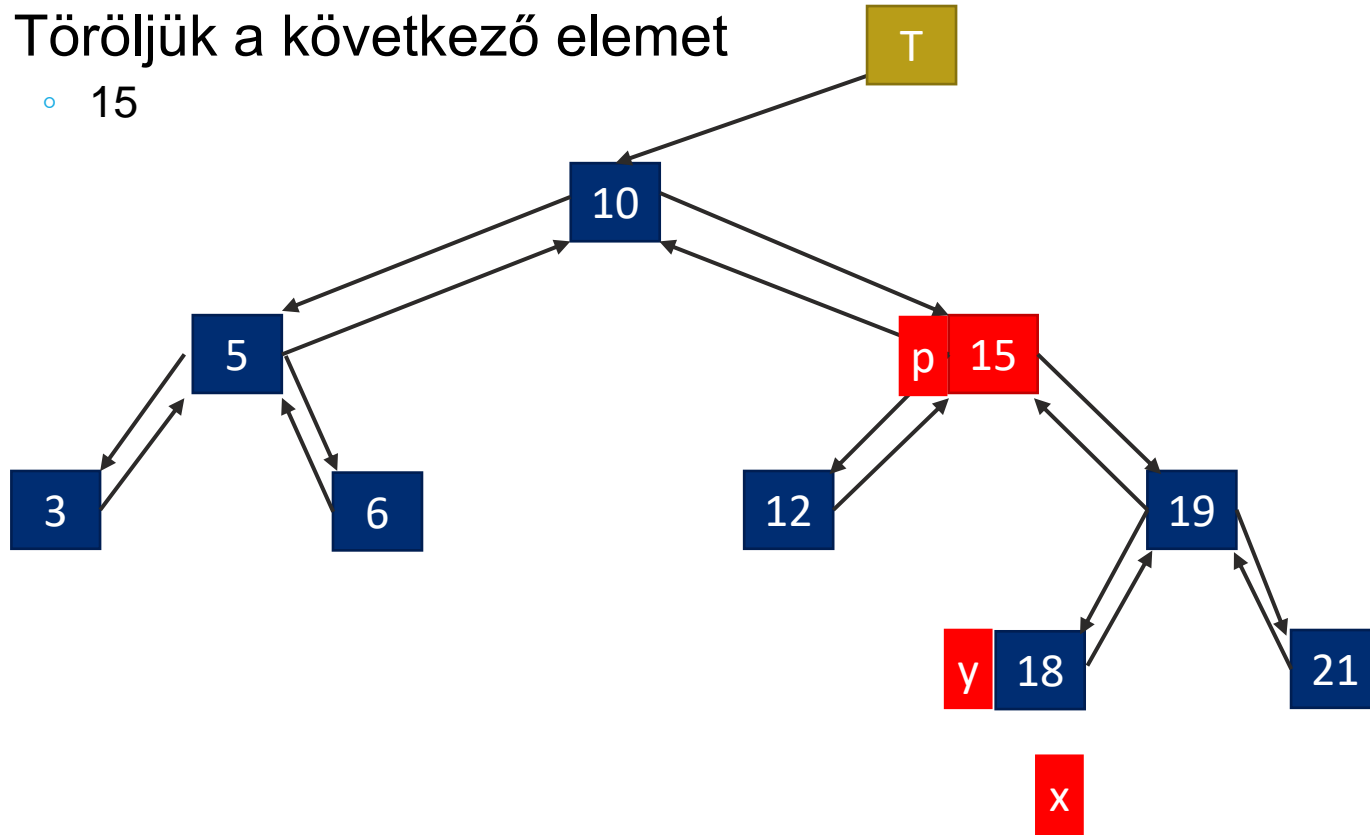
Fából-töröl (T,p)

```
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]
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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



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]
```

```
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
```

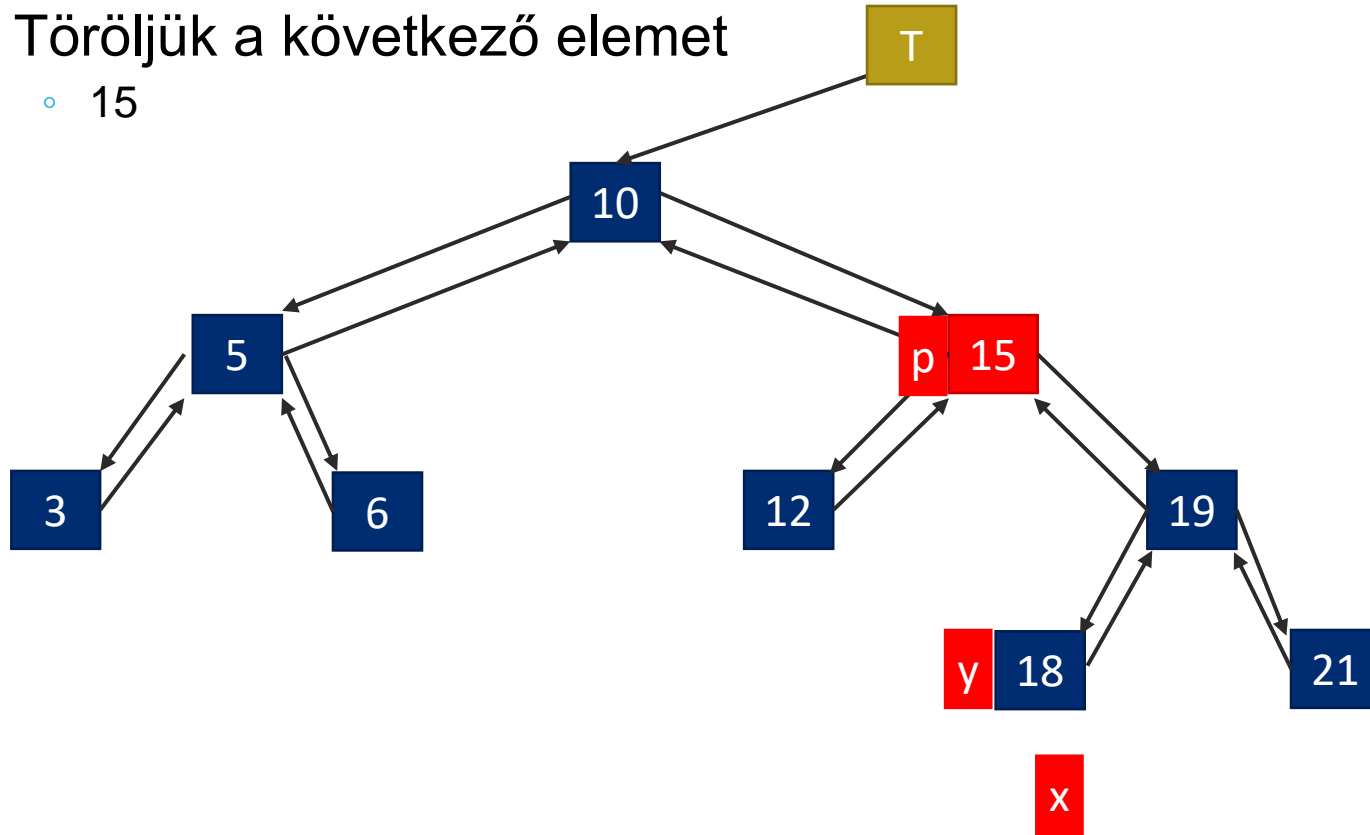
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



Fából-töröl (T,p)

if bal[p] = NIL vagy jobb[p] = NIL

then  $y \leftarrow p$

else  $y \leftarrow$  Fában-következő(T, p)

if bal[y]  $\neq$  NIL

then  $x \leftarrow$  bal[y]

else  $x \leftarrow$  jobb[y]

if  $x \neq$  NIL

then szülő[x]  $\leftarrow$  szülő[y]

if szülő[y] = NIL

then gyökér[T]  $\leftarrow$  x

else if  $y =$  bal[szülő[y]]

then bal[szülő[y]]  $\leftarrow$  x

else jobb[szülő[y]]  $\leftarrow$  x

if  $y \neq p$

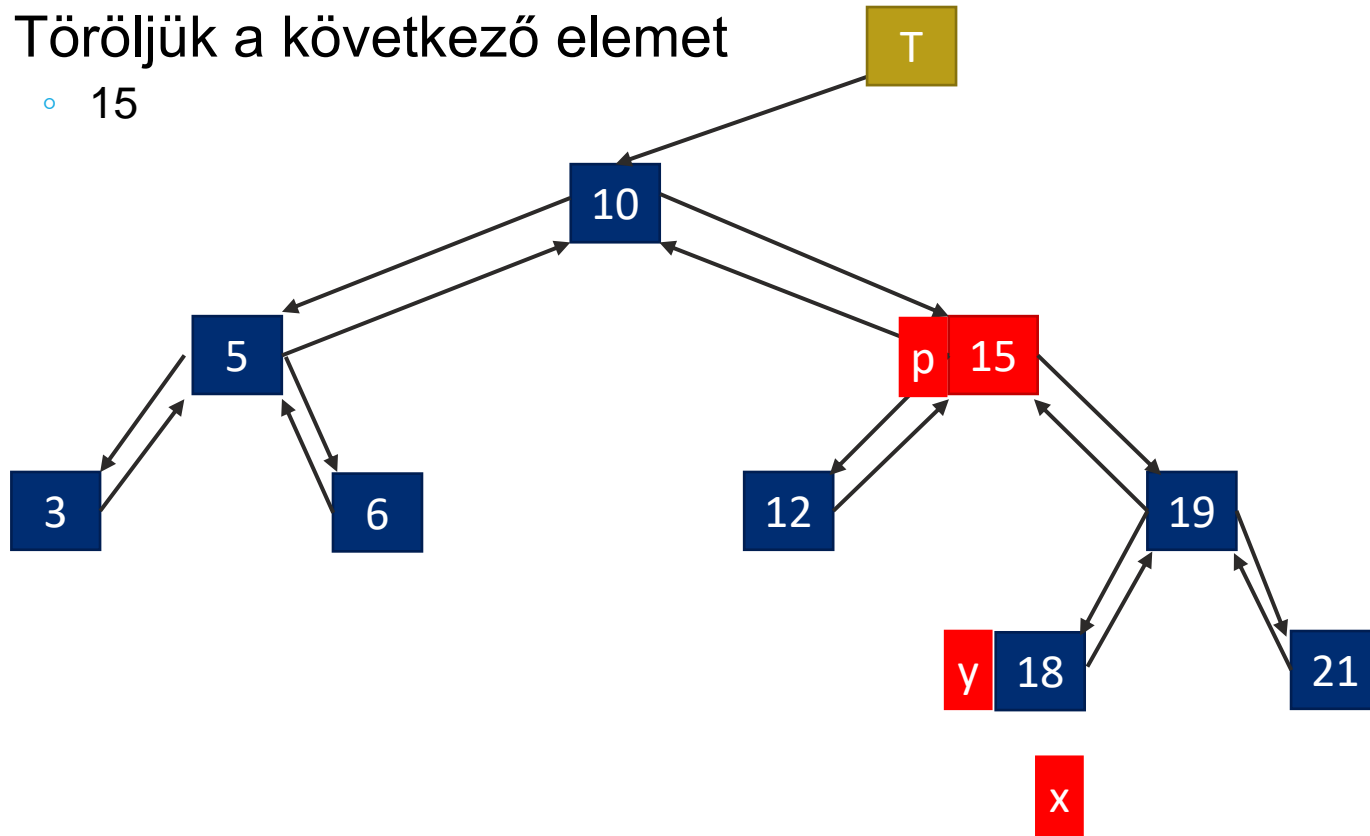
then kulcs[p]  $\leftarrow$  kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



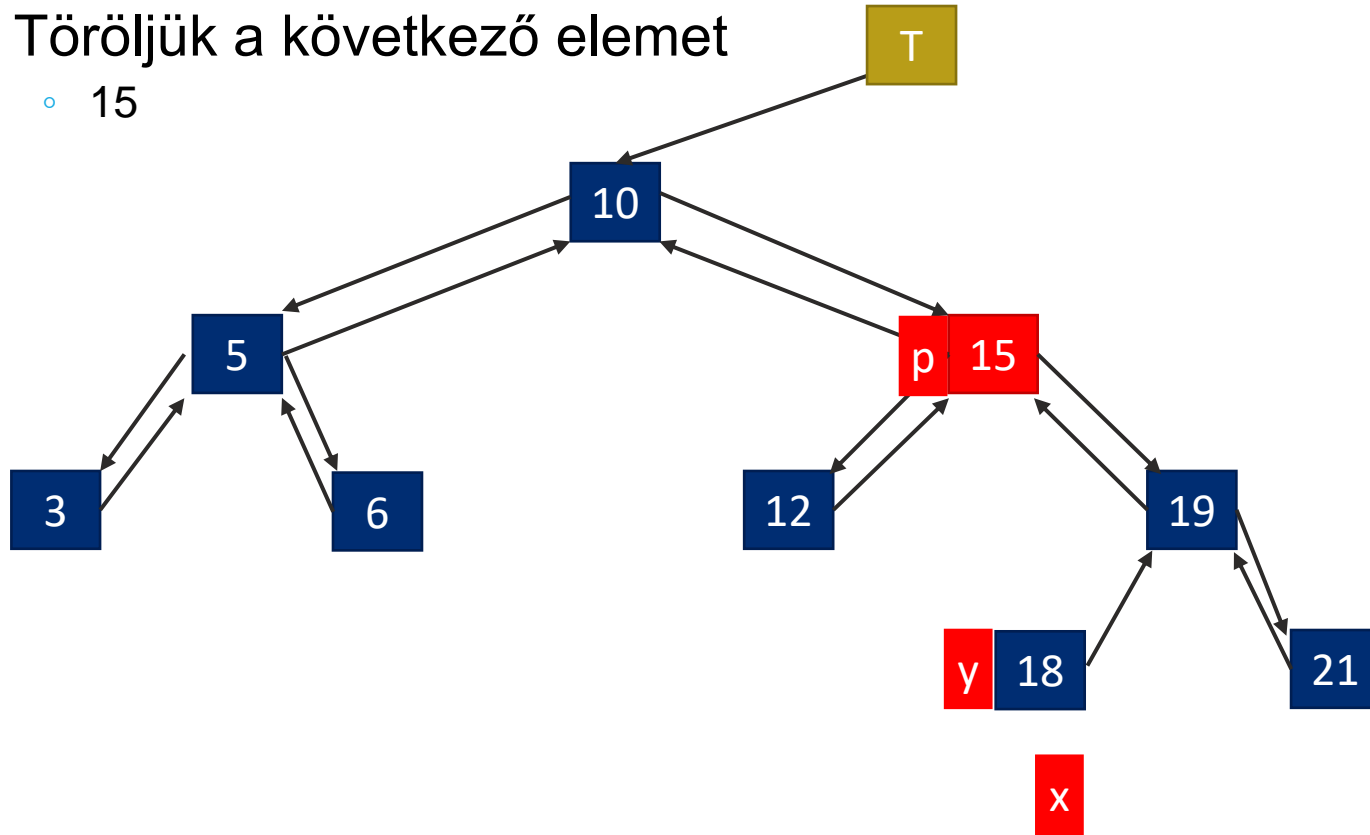
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



Fából-töröl (T,p)

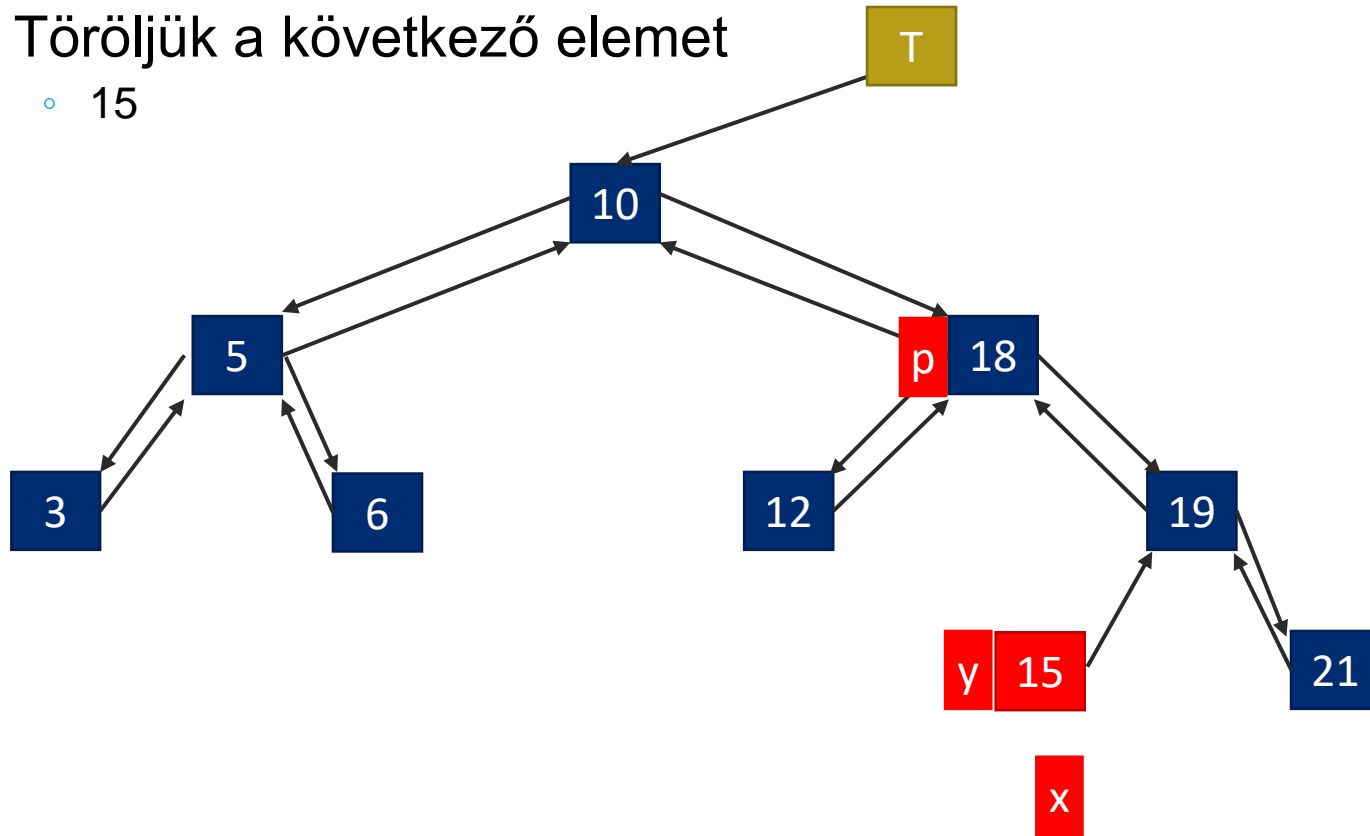
```
if bal[p] = NIL vagy jobb[p] = NIL
then y ← p
else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
then x ← bal[y]
else x ← jobb[y]
if x ≠ NIL
then szülő[x] ← szülő[y]
if szülő[y] = NIL
then gyökér[T] ← x
else if y = bal[szülő[y]]
then bal[szülő[y]] ← x
else jobb[szülő[y]] ← x
if y ≠ p
then kulcs[p] ←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



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



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]
```

```
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
```

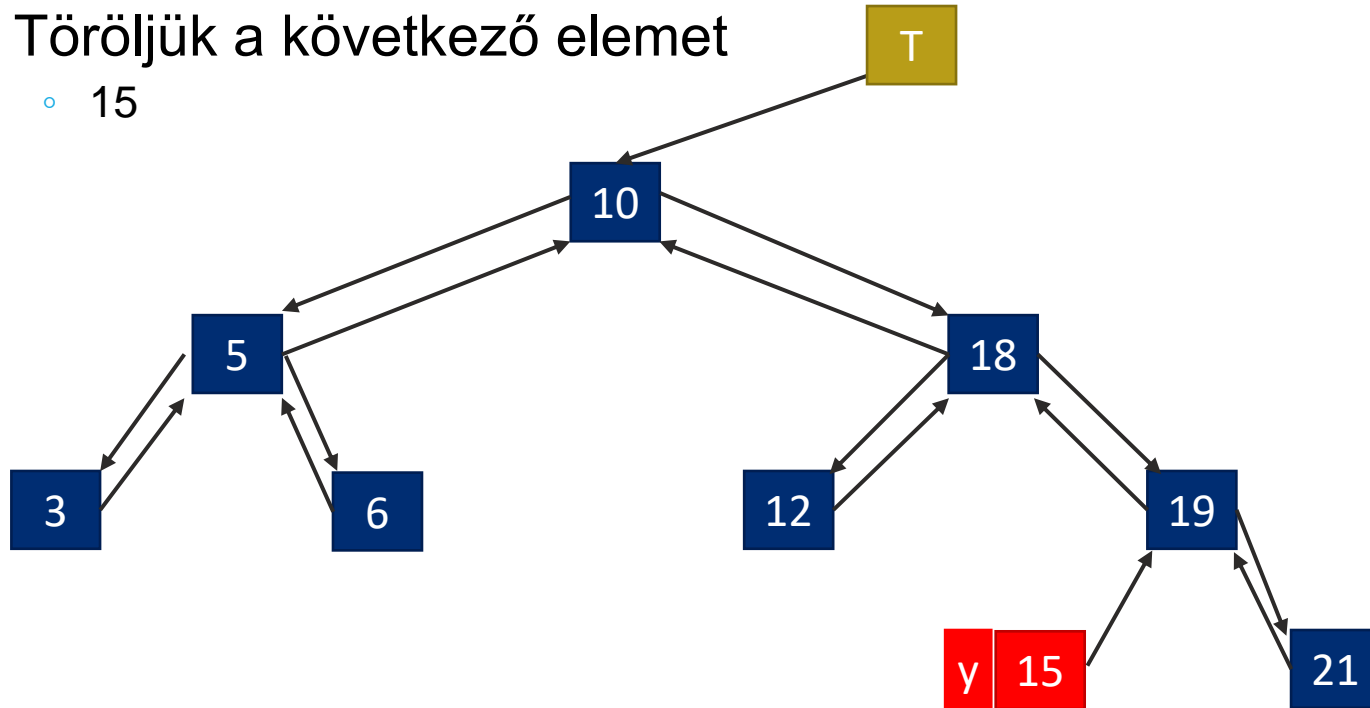
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 15



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]
```

```
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
```

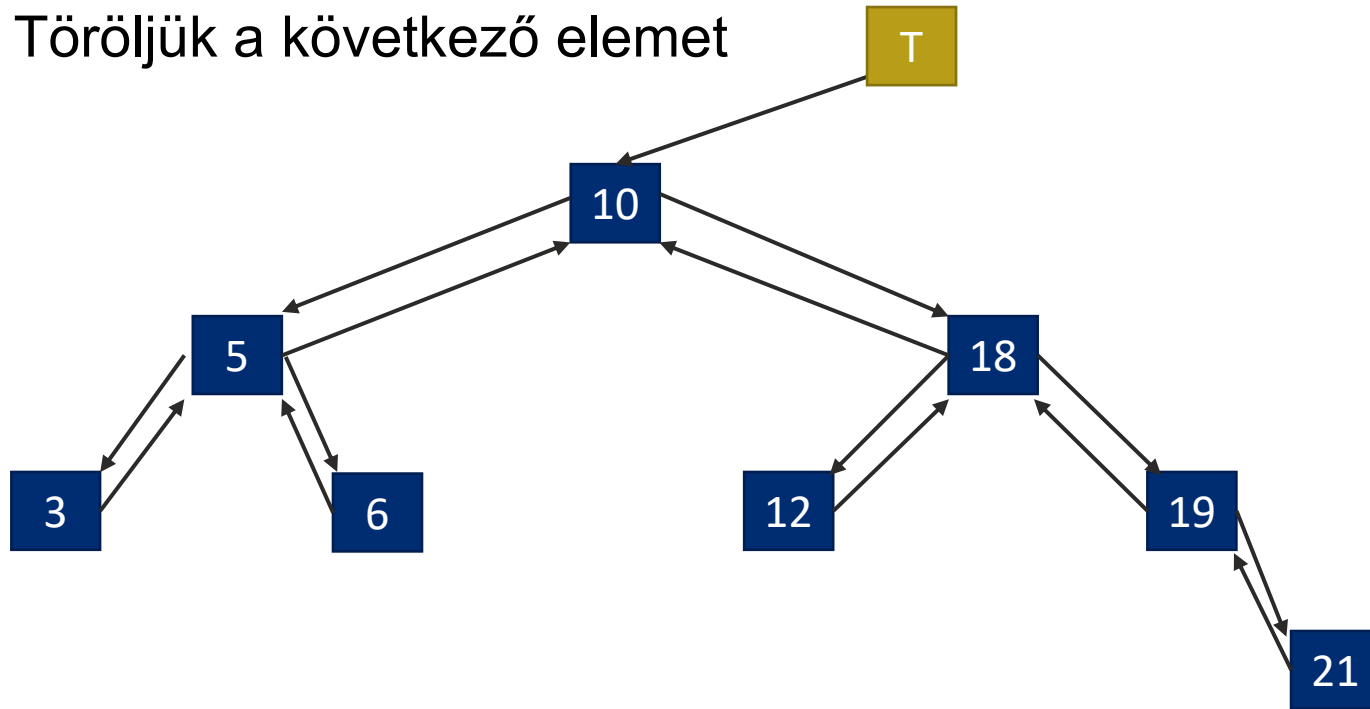
```
if y ≠ p
```

```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet



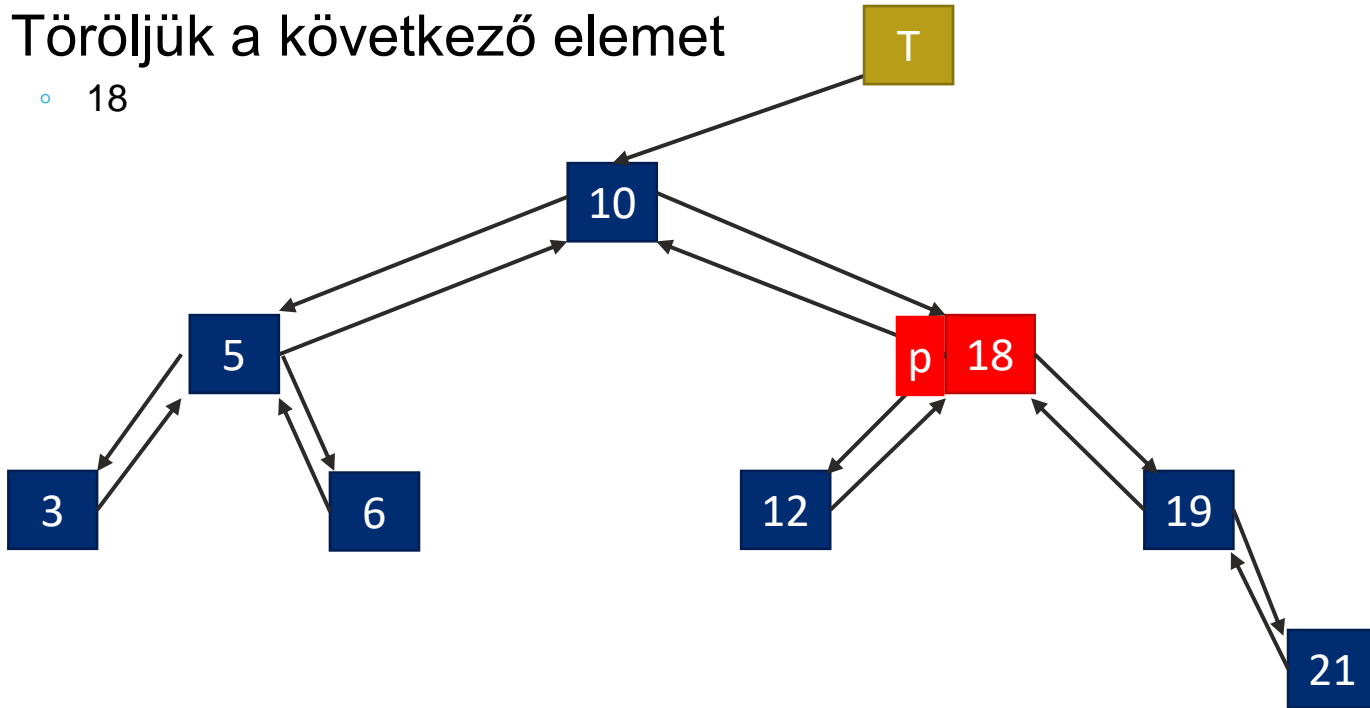
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

• 18



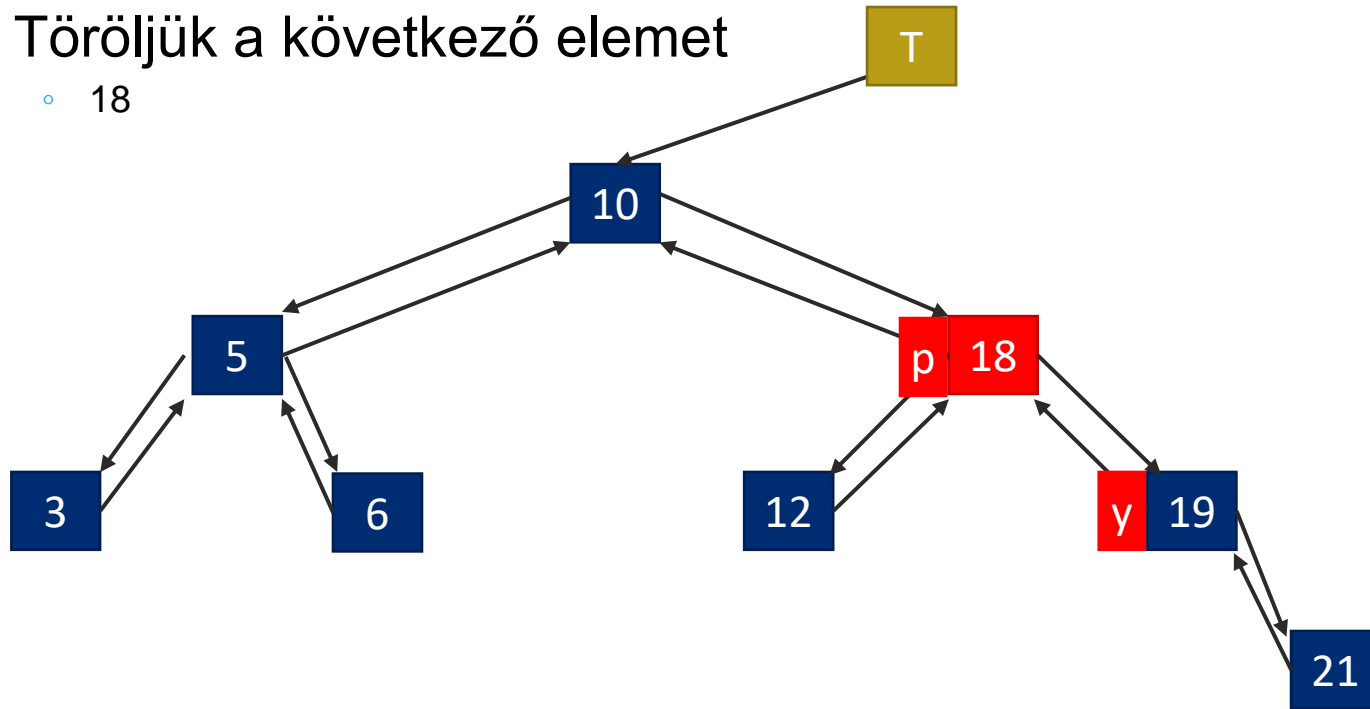
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 18



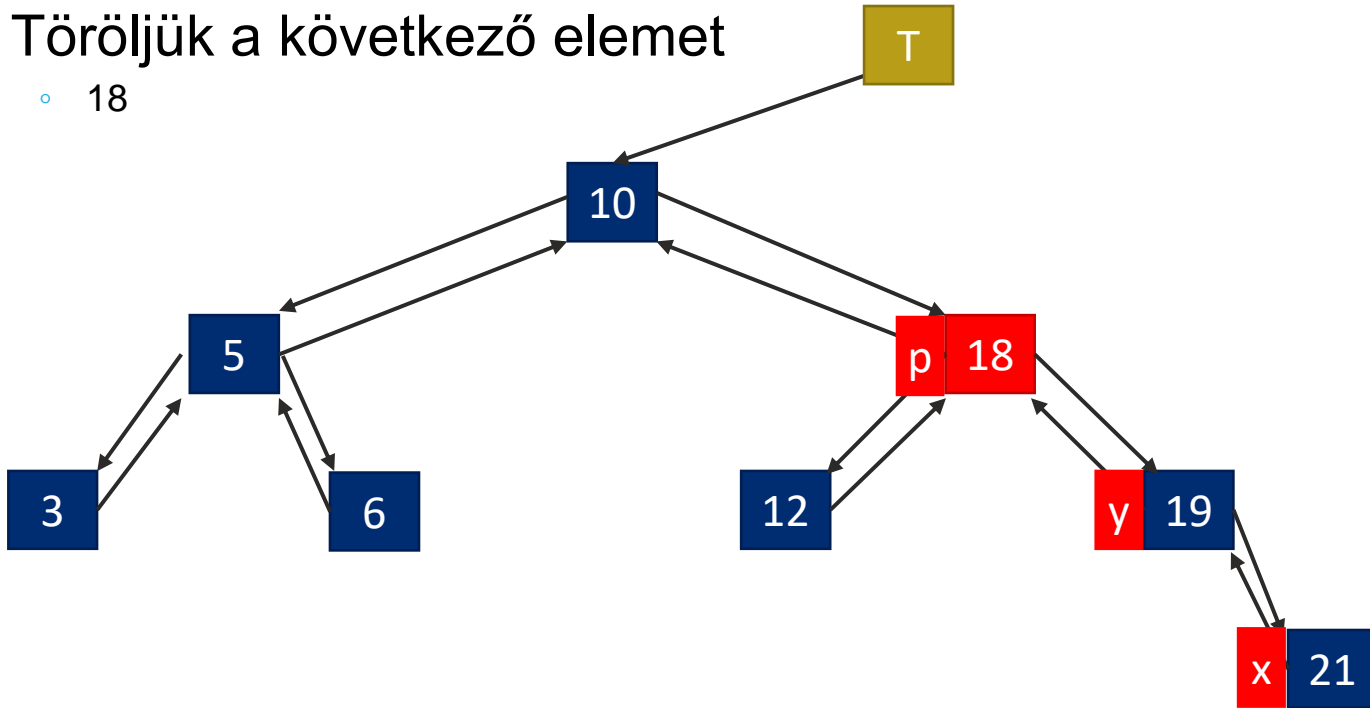
Fából-töröl (T,p)

```
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]
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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 18



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]

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

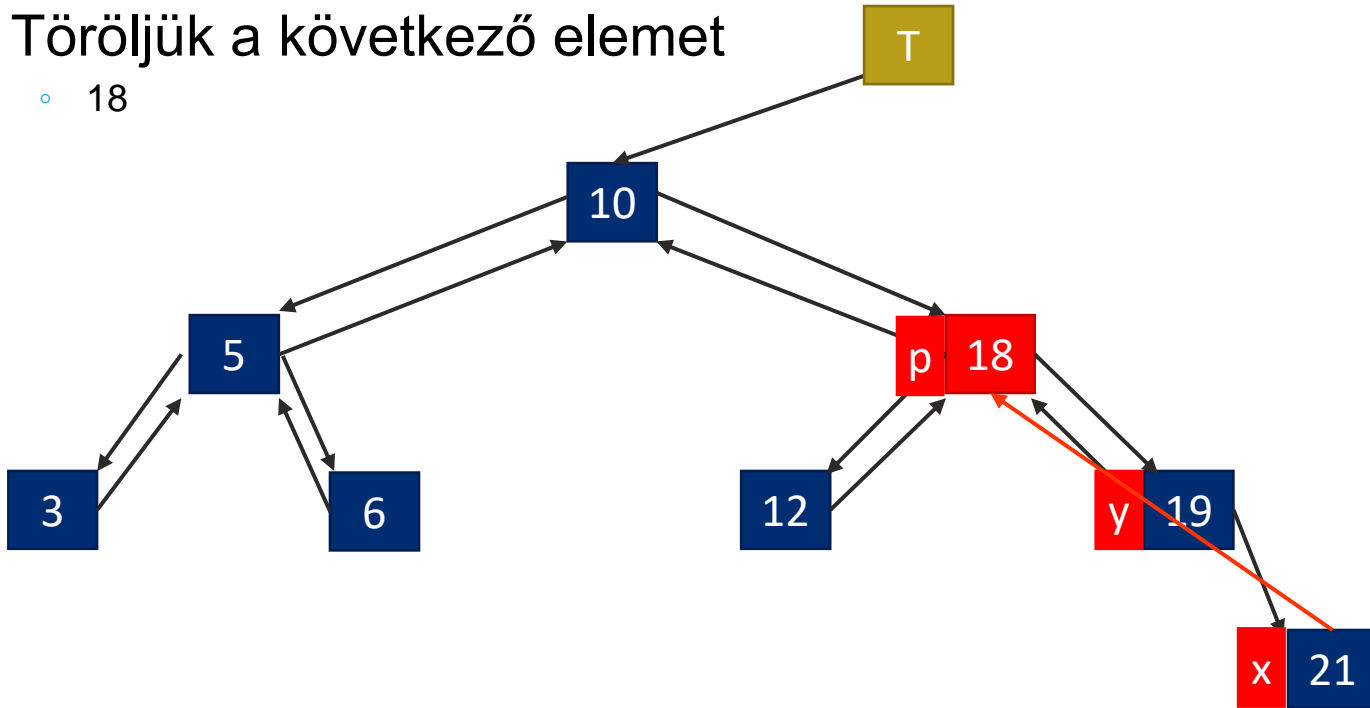
then kulcs[p] ← kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

• 18



Fából-töröl (T,p)

if bal[p] = NIL vagy jobb[p] = NIL

then  $y \leftarrow p$

else  $y \leftarrow$  Fában-következő(T, p)

if bal[y]  $\neq$  NIL

then  $x \leftarrow$  bal[y]

else  $x \leftarrow$  jobb[y]

if  $x \neq$  NIL

then  $szülő[x] \leftarrow szülő[y]$

if  $szülő[y] =$  NIL

then  $gyökér[T] \leftarrow x$

else if  $y =$  bal[ $szülő[y]$ ]

then bal[ $szülő[y]$ ]  $\leftarrow x$

else jobb[ $szülő[y]$ ]  $\leftarrow x$

if  $y \neq p$

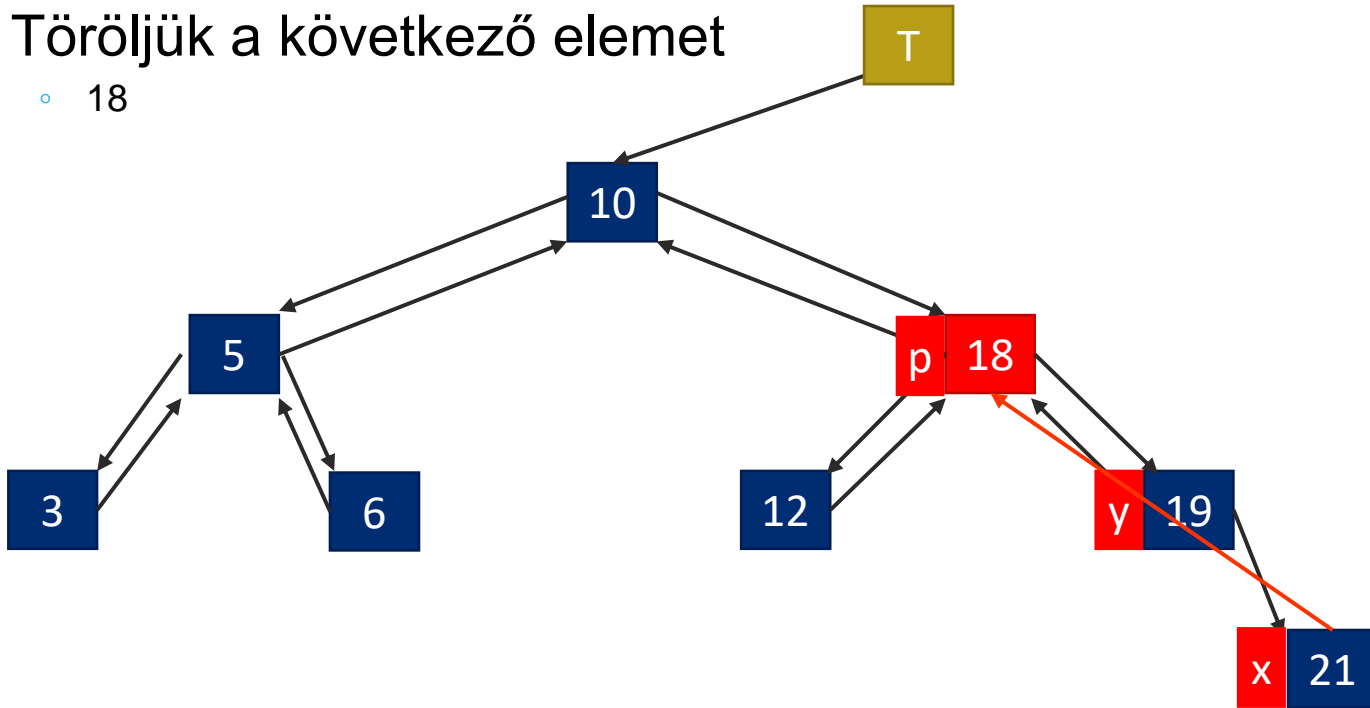
then kulcs[p]  $\leftarrow$  kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

• 18



Fából-töröl (T,p)

if bal[p] = NIL vagy jobb[p] = NIL

then  $y \leftarrow p$

else  $y \leftarrow$  Fában-következő(T, p)

if bal[y]  $\neq$  NIL

then  $x \leftarrow$  bal[y]

else  $x \leftarrow$  jobb[y]

if  $x \neq$  NIL

then szülő[x]  $\leftarrow$  szülő[y]

if szülő[y] = NIL

then gyökér[T]  $\leftarrow$  x

else if  $y =$  bal[szülő[y]]

then bal[szülő[y]]  $\leftarrow$  x

else jobb[szülő[y]]  $\leftarrow$  x

if  $y \neq p$

then kulcs[p]  $\leftarrow$  kulcs[y]

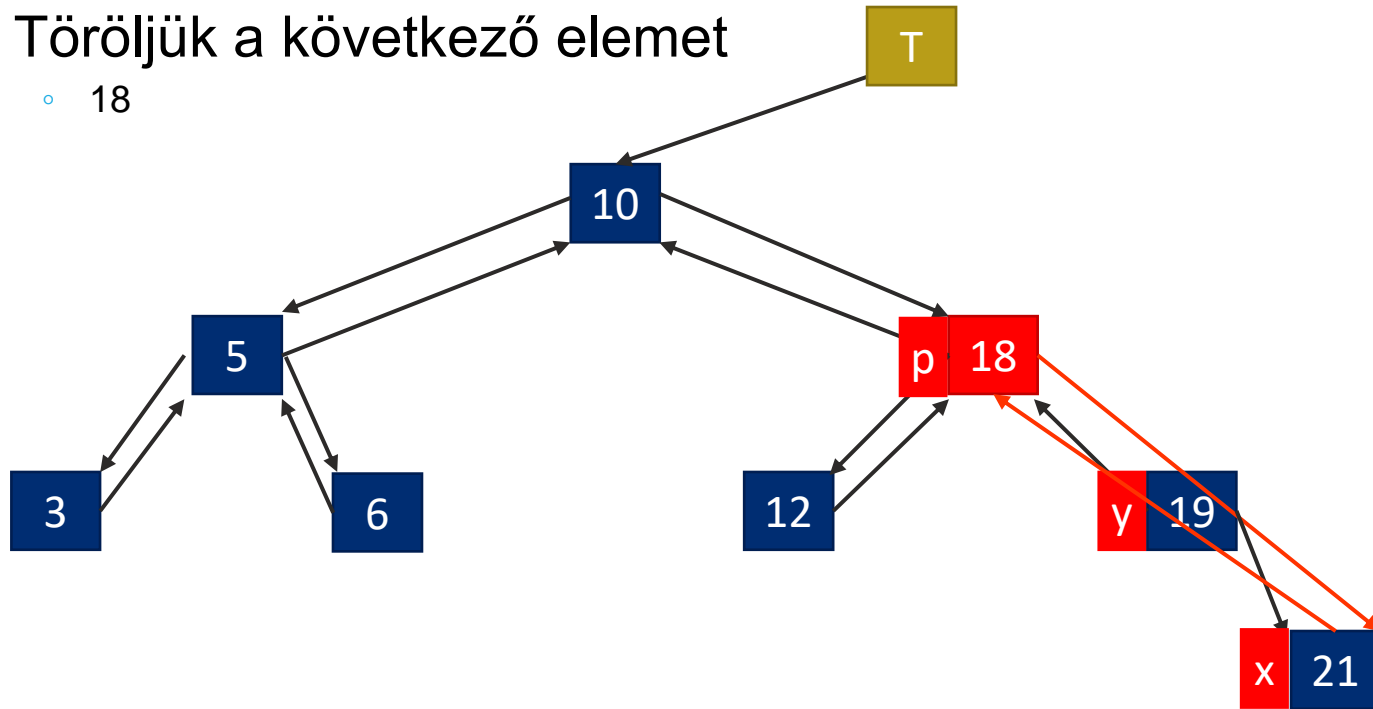
return y



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

• 18



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]
```

```
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
```

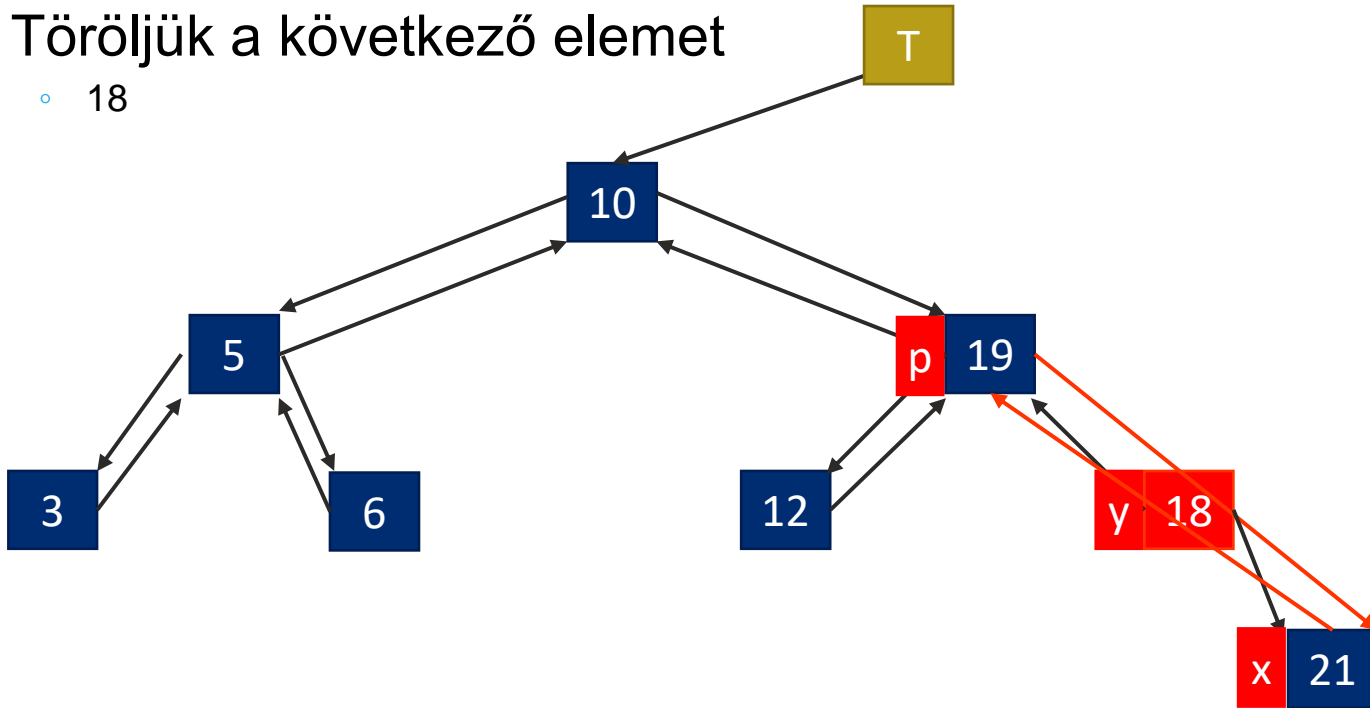
```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

• 18



Fából-töröl (T,p)

if bal[p] = NIL vagy jobb[p] = NIL

then  $y \leftarrow p$

else  $y \leftarrow$  Fában-következő(T, p)

if bal[y]  $\neq$  NIL

then  $x \leftarrow$  bal[y]

else  $x \leftarrow$  jobb[y]

if  $x \neq$  NIL

then szülő[x]  $\leftarrow$  szülő[y]

if szülő[y] = NIL

then gyökér[T]  $\leftarrow$  x

else if  $y =$  bal[szülő[y]]

then bal[szülő[y]]  $\leftarrow$  x

else jobb[szülő[y]]  $\leftarrow$  x

if  $y \neq p$

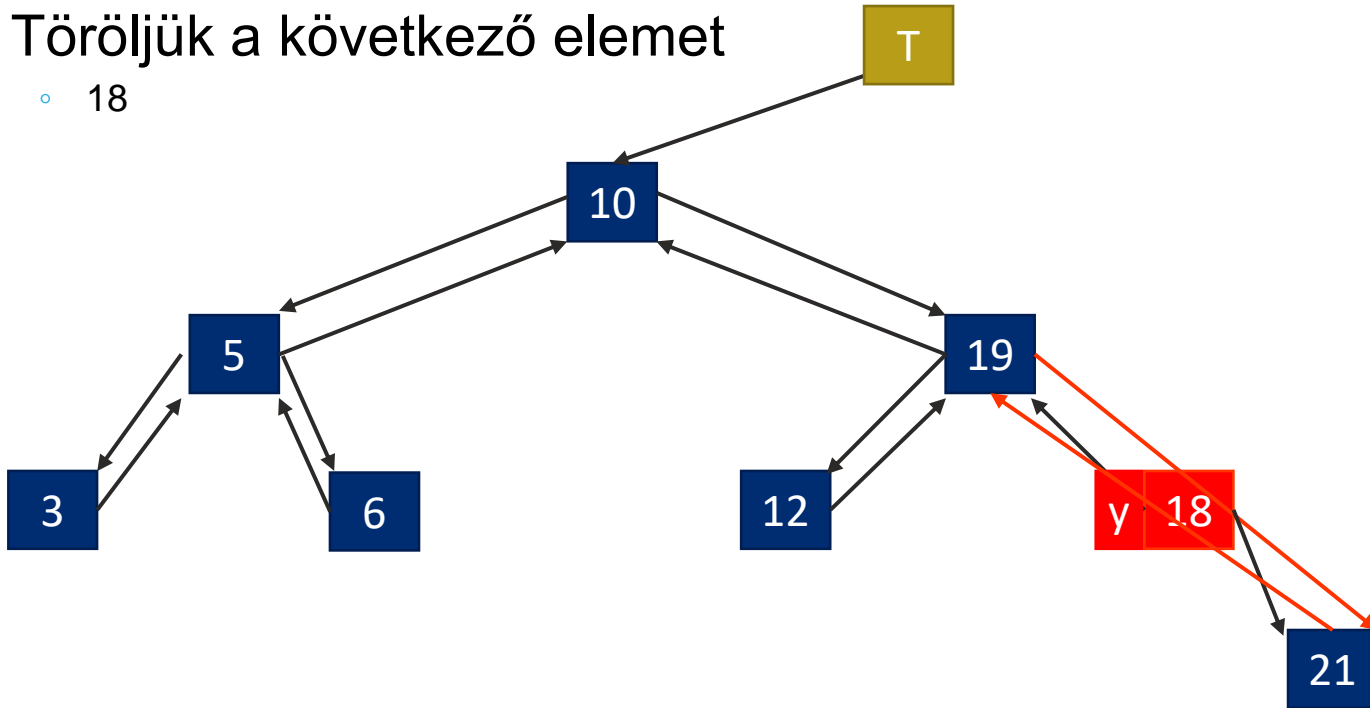
then kulcs[p]  $\leftarrow$  kulcs[y]

return y

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

• 18



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]
```

```
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
```

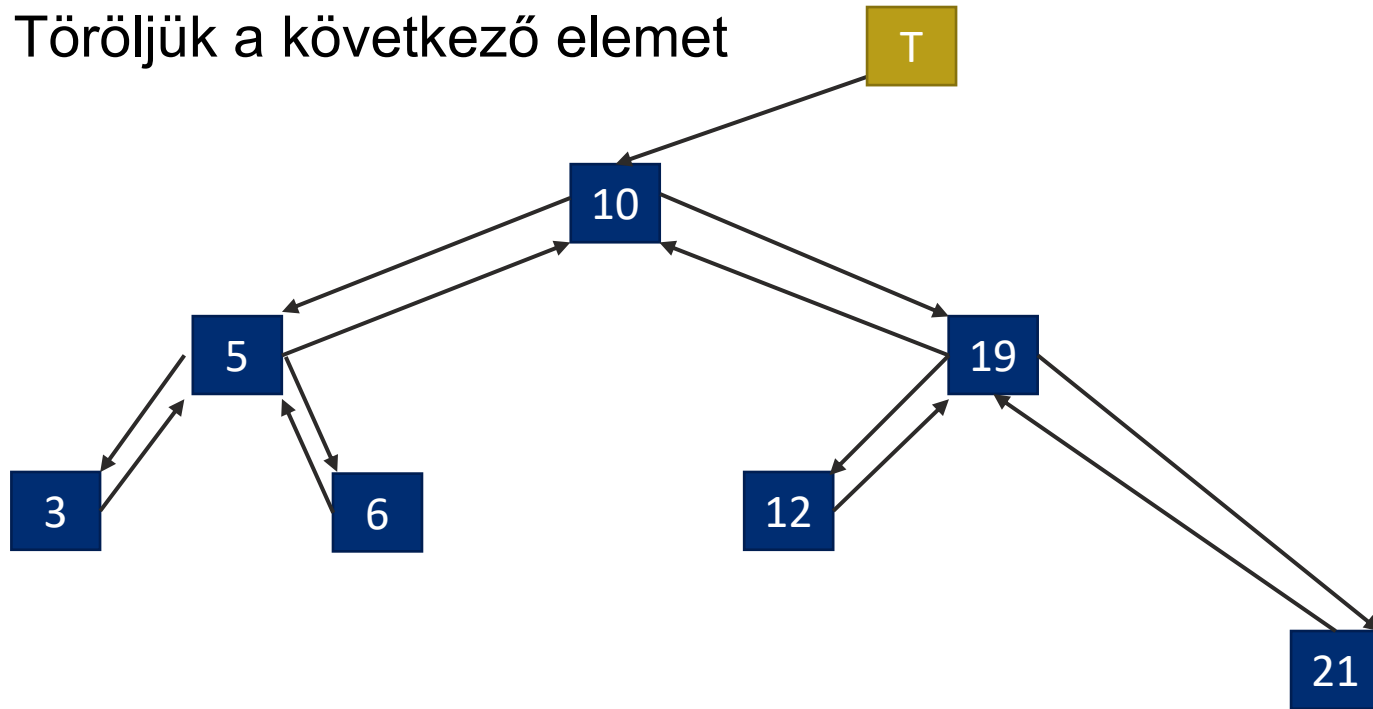
```
if y ≠ p
```

```
then kulcs[p] ← kulcs[y]
```

```
return y
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

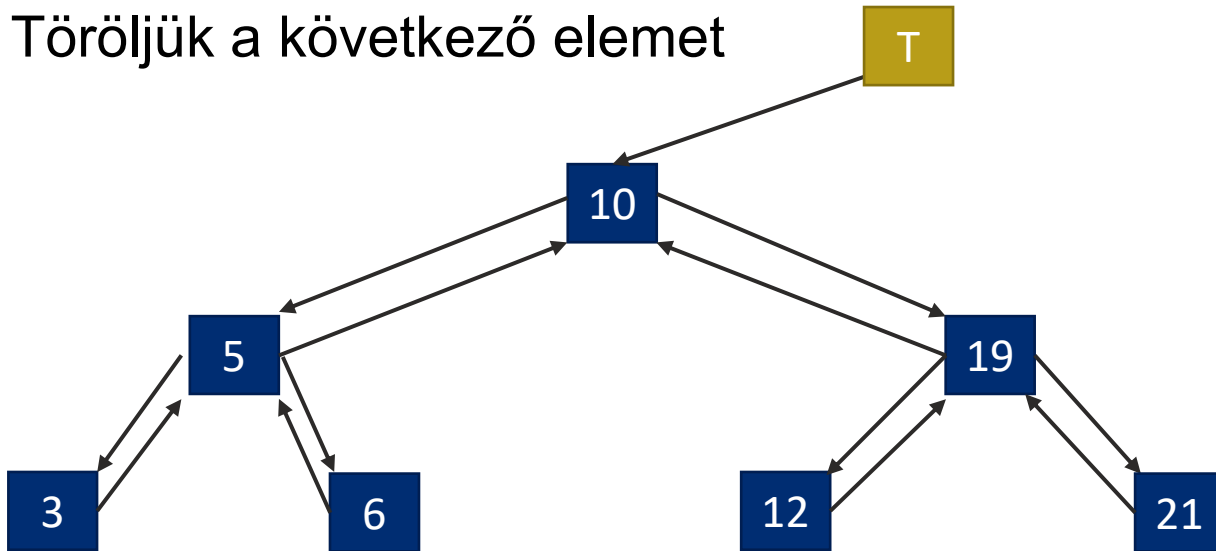


Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet



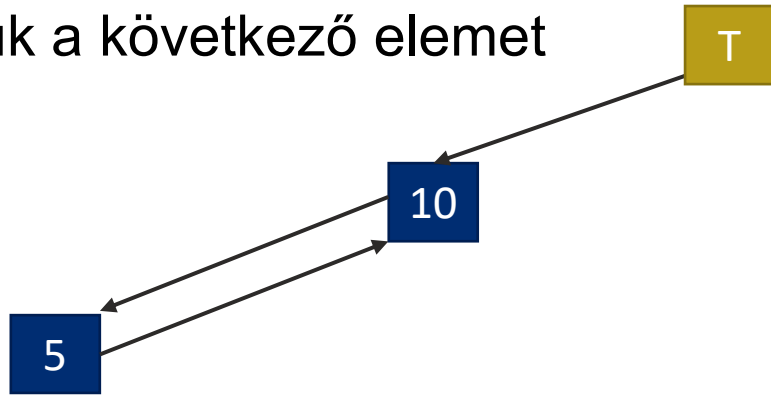
Fából-töröl (T,p)

```
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]
  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
```

További elemeket is törölünk.

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet



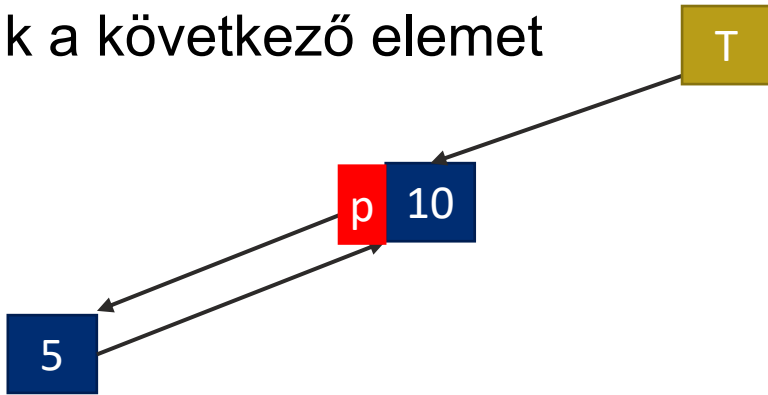
```
Fából-töröl (T,p)
if bal[p] = NIL vagy jobb[p] = NIL
  then y ← p
  else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
  then x ← bal[y]
  else x ← jobb[y]
if x ≠ NIL
  then szülő[x] ← szülő[y]
if szülő[y] = NIL
  then gyökér[T] ← x
  else if y = bal[szülő[y]]
    then bal[szülő[y]] ← x
    else jobb[szülő[y]]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

További elemeket is törölünk.

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10



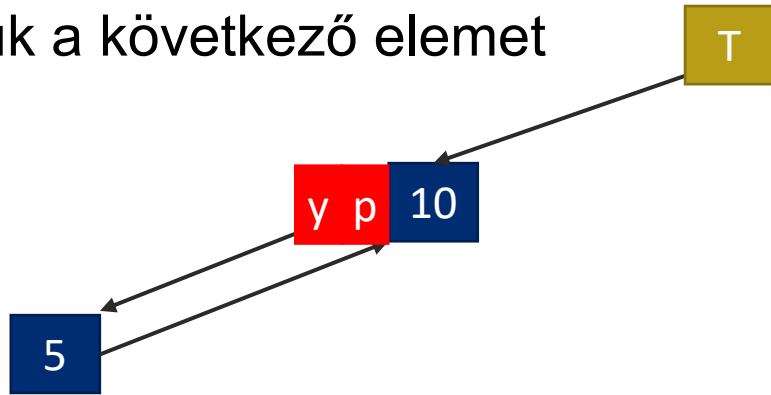
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10



Fából-töröl (T,p)

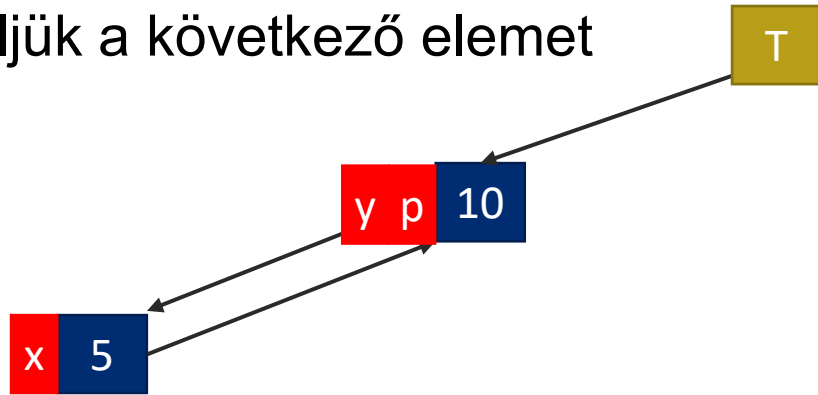
```
if bal[p] = NIL vagy jobb[p] = NIL
  then y ← p
  else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
  then x ← bal[y]
  else x ← jobb[y]
if x ≠ NIL
  then szülő[x] ← szülő[y]
if szülő[y] = NIL
  then gyökér[T] ← x
  else if y = bal[szülő[y]]
    then bal[szülő[y]] ← x
    else jobb[szülő[y]] ← x
if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



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10



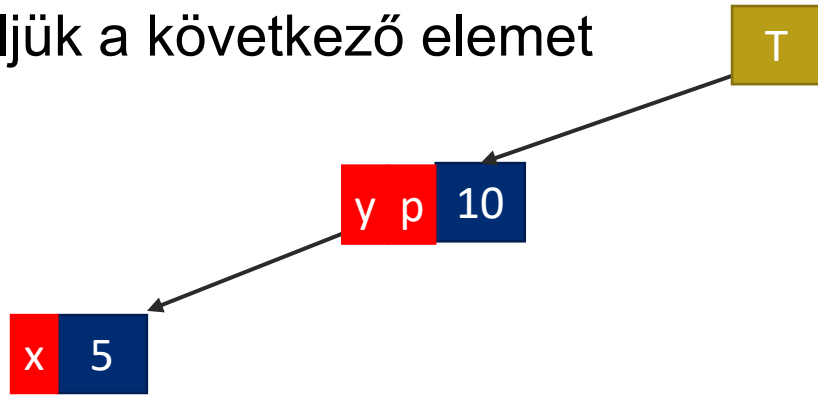
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10



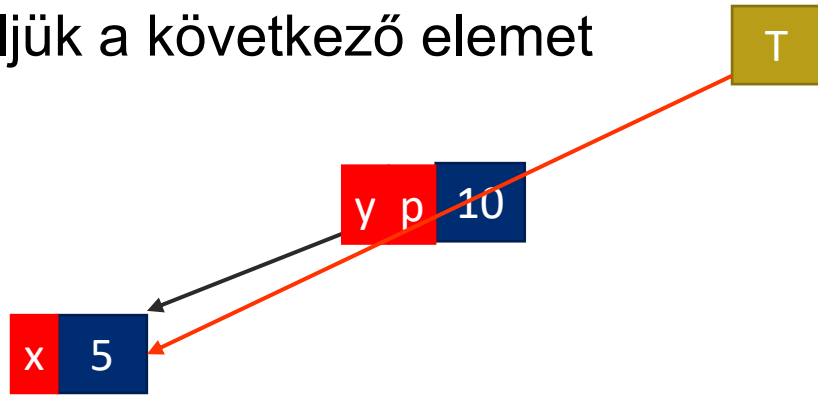
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10



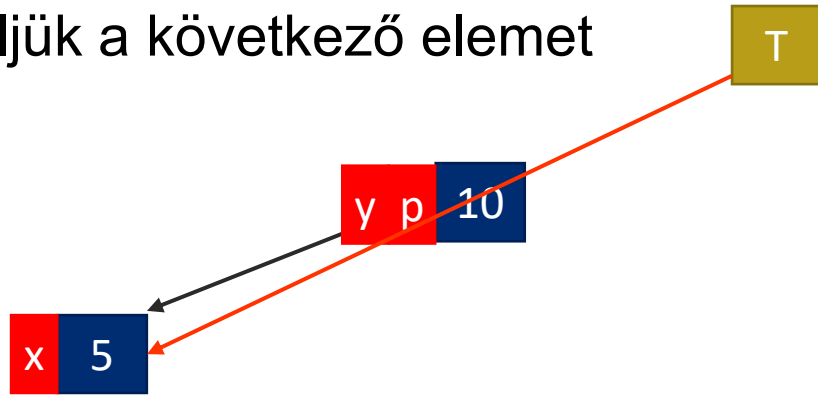
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10



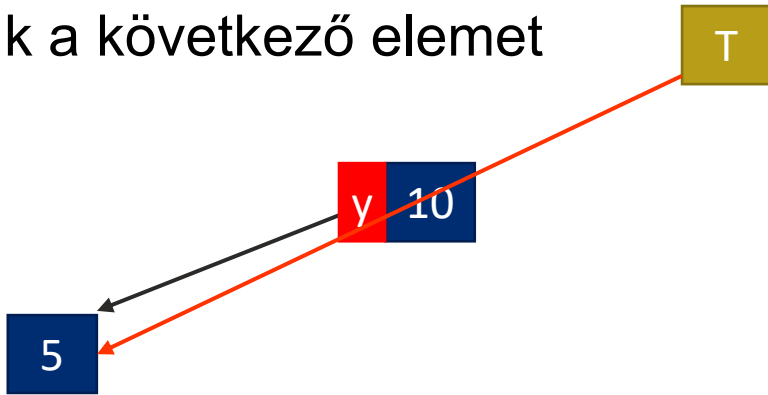
Fából-töröl (T,p)

```
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]
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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 10

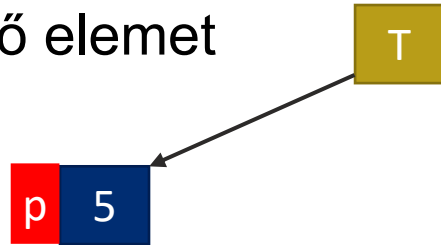


Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

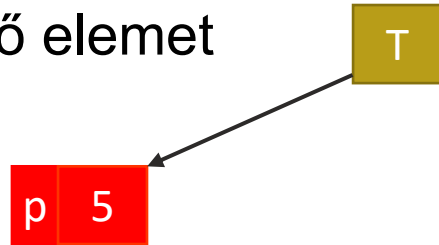


```
Fából-töröl (T,p)
if bal[p] = NIL vagy jobb[p] = NIL
  then y ← p
  else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
  then x ← bal[y]
  else x ← jobb[y]
if x ≠ NIL
  then szülő[x] ← szülő[y]
if szülő[y] = NIL
  then gyökér[T] ← x
  else if y = bal[szülő[y]]
    then bal[szülő[y]] ← x
    else jobb[szülő[y]]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

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 5

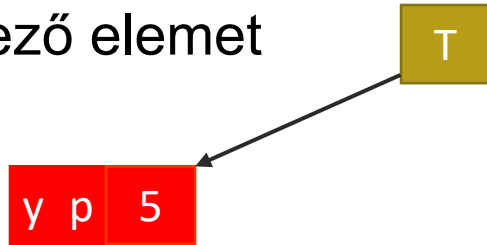


```
Fából-töröl (T,p)
if bal[p] = NIL vagy jobb[p] = NIL
  then y ← p
  else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
  then x ← bal[y]
  else x ← jobb[y]
if x ≠ NIL
  then szülő[x] ← szülő[y]
if szülő[y] = NIL
  then gyökér[T] ← x
  else if y = bal[szülő[y]]
    then bal[szülő[y]] ← x
    else jobb[szülő[y]]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

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 5



Fából-töröl (T,p)

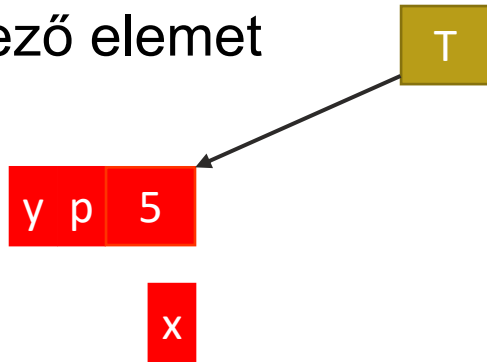
```
if bal[p] = NIL vagy jobb[p] = NIL
then y ← p
else y ← Fában-következő(T, p)
if bal[y] ≠ NIL
then x ← bal[y]
else x ← jobb[y]
if x ≠ NIL
then szülő[x] ← szülő[y]
if szülő[y] = NIL
then gyökér[T] ← x
else if y = bal[szülő[y]]
then bal[szülő[y]] ← x
else jobb[szülő[y]] ← x
if y ≠ p
then kulcs[p] ←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



# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 5



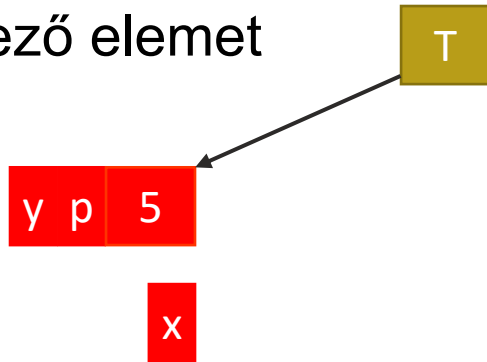
Fából-töröl (T,p)

```
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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

- 5



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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

◦ 5

y p 5

x

T

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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

◦ 5

y p 5

x

T

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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

◦ 5

y p 5

x

T

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]
  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
```

# Bináris keresési fa – Elem törlése

Töröljük a következő elemet

T

```
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]
  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
```

# Keresés hatékonysága

Következő téma