

Algoritmusok gyakorlat

8. gyakorlat, október 26.

8. gyakorlat

Fa = összefüggő, körmentes gráf

Bármely két csúcsát *pontosan egy* út köti össze

A fa elsőfokú csúcsait **levél**nek hívjuk. Egy nem levél csúcs a fában **belső csúcs**.

Hány éle van egy fának, ha n csúcsa van?

8. gyakorlat

Gyökeres fa: van egy kitüntetett csúcsa, a gyökér

Bináris fa: gyökeres fa, ahol minden csúcsnak legfeljebb két gyereke van

Fa ábrázolások: gyerek éllista; első fiú, apa, testvér; speciális: bináris fa

8. gyakorlat

Bináris keresőfa: legyen x egy adott bináris keresőfa egy tetszőleges csúcsa. Ha y az x baloldali részfájának egy csúcsa, akkor **$\text{kulcs}[y] \leq \text{kulcs}[x]$.**

Ha y az x jobboldali részfájának egy csúcsa, akkor

$\text{kulcs}[x] \leq \text{kulcs}[y]$.

8. gyakorlat

Elemi műveletek keresőfákban (példák később):

Keres

Min/max

Következő (nagyság szerint)

Beszúr

Töröl (levél, egy gyerek, két gyerek)

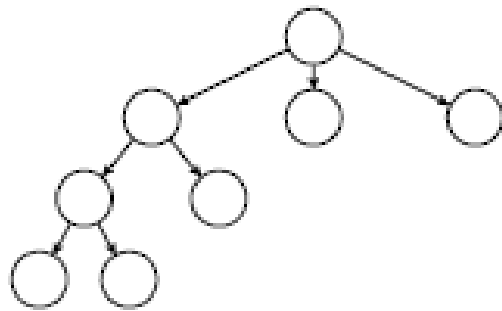
Fontos: fa magassága!!!

8. gyakorlat

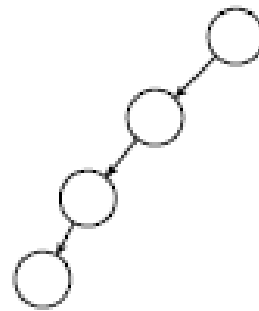
- Teljes bináris fa: olyan bináris fa, ahol minden szint teljesen ki van töltve
- Majdnem teljes bináris fa: olyan bináris fa, ahol maximum a legalsó szint nincs teljesen kitöltve, csak balról jobbra haladva kitöltött néhány elemig
- Kiegyensúlyozott bináris fa: olyan bináris fa, ahol minden csúcs bal és jobb oldali részfáinak magassága maximum eggyel tér el.

8. gyakorlat

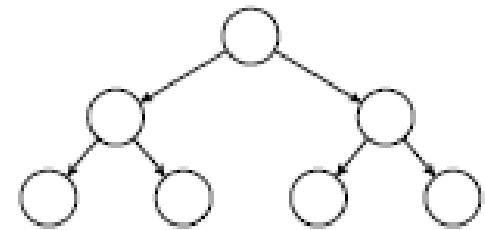
1. Feladat Döntsük el a következő fákról, hogy mely tulajdonság igaz rájuk a következők közül: fa, bináris fa, teljes bináris fa, majdnem teljes bináris fa, kiegyensúlyozott bináris fa!



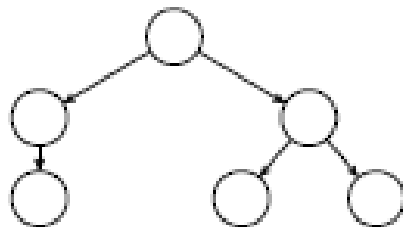
(a)



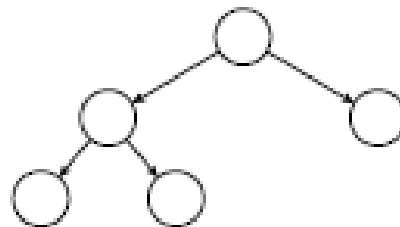
(b)



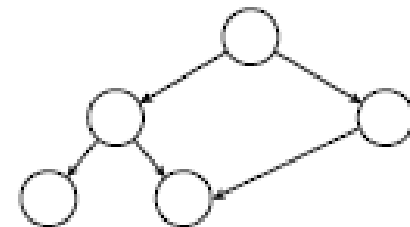
(c)



(d)



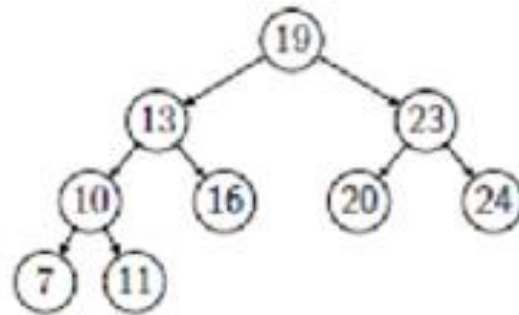
(e)



(f)

8. gyakorlat

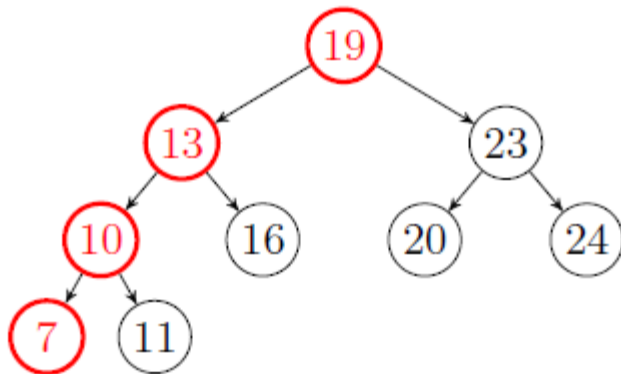
2. Feladat Keressük meg a következő keresőfában az alábbi elemeket: 5, 16, 24.



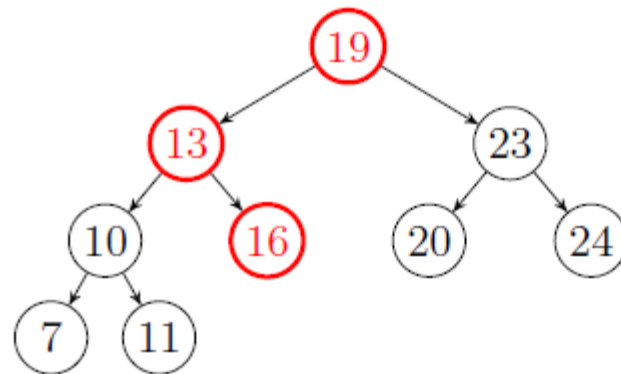
8. gyakorlat

```
public Node search(Node root, int key)
{
    if (root==null || root.key==key)
        return root;
    if (root.key > key)
        return search(root.left, key);
    return search(root.right, key);
}
```

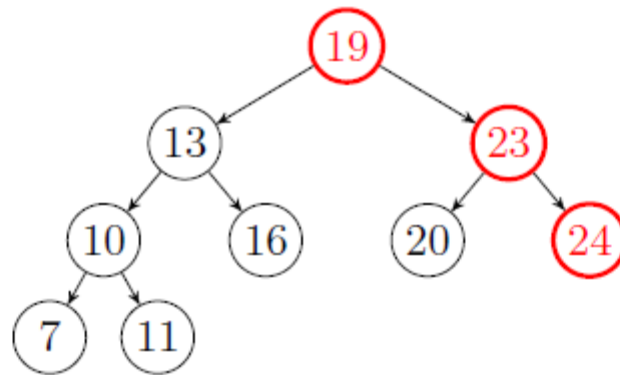
8. gyakorlat



(a) search(5)



(b) search(16)



(c) search(24)

8. gyakorlat

Előző fa minimális, maximális eleme? Melyik elem következik 19 után? Mi van előtte?

8. gyakorlat

```
Node minValue(Node node)
```

```
{
    Node current = node;
    while (current.left != null) {
        current = current.left;
    }
    return current;
}
```

```
Node successor(Node root, Node n)
```

```
{
    if (n.right != null) {
        return minValue(n.right);
    }
    Node p = n.parent;
    while (p != null && n == p.right) {
        n = p;
        p = p.parent;
    }
    return p;
}
```

```
Node maxValue(Node node)
```

```
{
    Node current = node;
    while (current.right != null) {
        current = current.right;
    }
    return current;
}
```

```
Node predecessor(Node root, Node n)
```

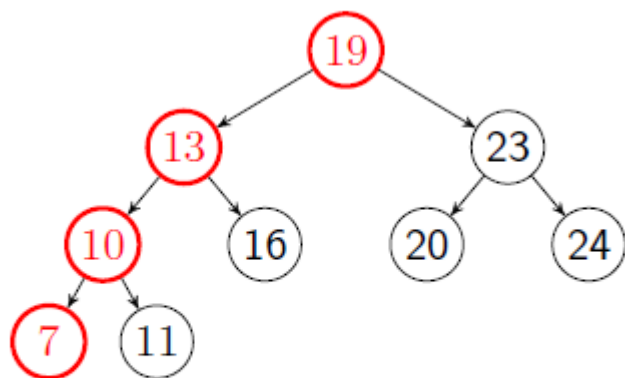
```
{
    if (n.left != null) {
        return maxValue(n.left);
    }
    Node p = n.parent;
    while (p != null && n == p.left) {
        n = p;
        p = p.parent;
    }
    return p;
}
```

8. gyakorlat

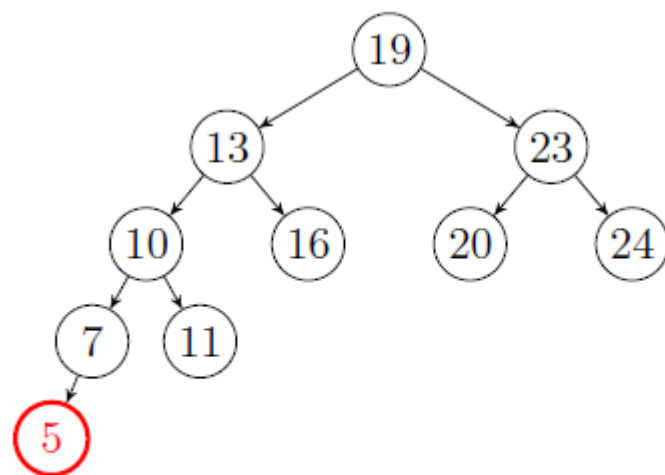
3.Feladat: Szúrjuk be az előző keresőfába az 5 és 22 értékeket

8. gyakorlat

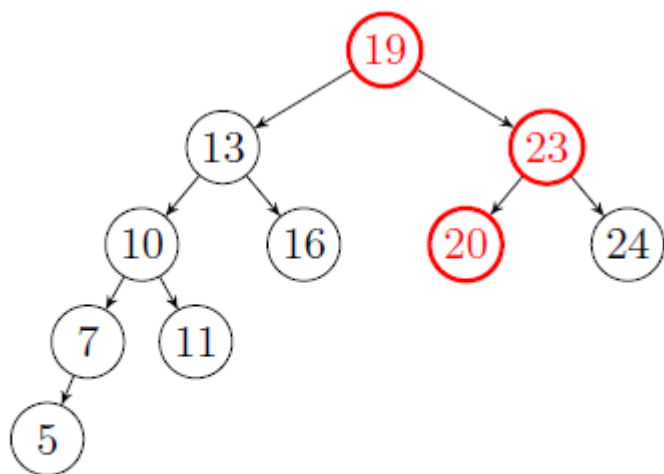
```
Node insert(Node root, int key) {  
    if (root == null) {  
        root = new Node(key);  
        return root;  
    }  
    if (key < root.key)  
        root.left = insert(root.left, key);  
    else if (key > root.key)  
        root.right = insert(root.right, key);  
    return root;  
}
```



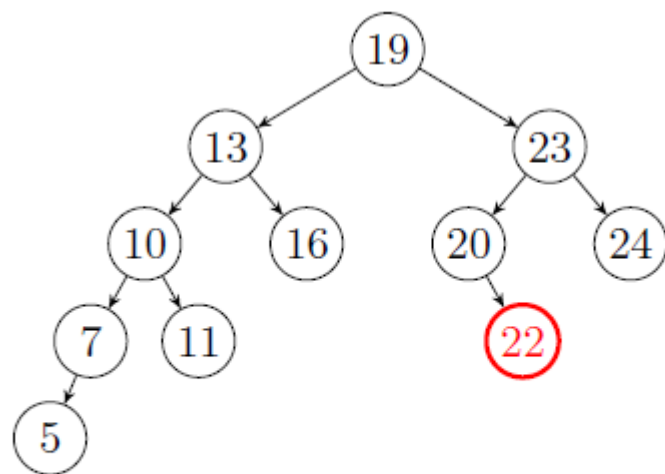
(a) search(5)



(b) new node()



(c) search(22)



(d) new node()

8. gyakorlat

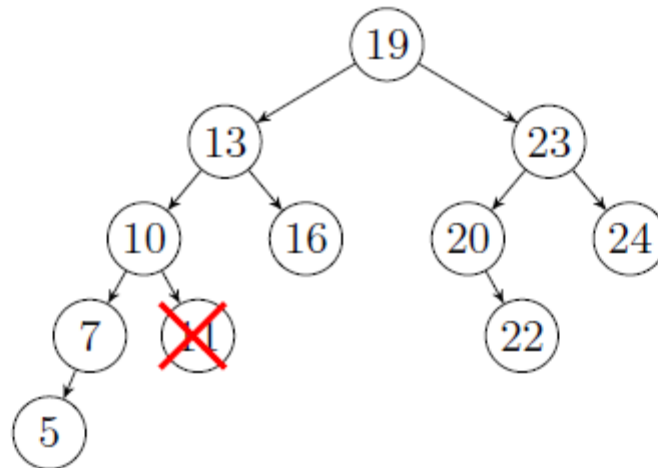
4.Feladat: Töröljük az előző feladatban beszúrások után kapott fából a 11, 7, 19 elemeket

8. gyakorlat

- Három esetet különböztetünk meg az x csúcs törlésekor:
- 1. Ha x -nek nincs gyereke, töröljük és a szülője mutatóját null-ra állítjuk.
- 2. Ha x -nek pontosan egy gyereke van, legyen ez c , akkor felemeljük az x helyére (x szülőjének a gyereke c , c szülője az x szülője lesz).
- 3. Ha az x -nek két gyereke van akkor megkeressük az x közvetlen rákövetkezőjét z -t, és az ő kulcsát tesszük x helyére a fában. Ezután az eredeti z csúcsot töröljük, neki biztosan nincs bal gyereke.

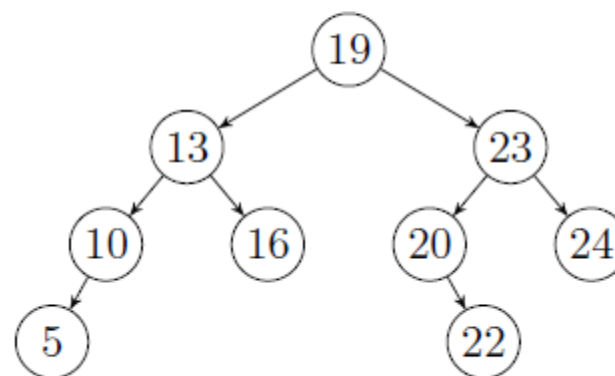
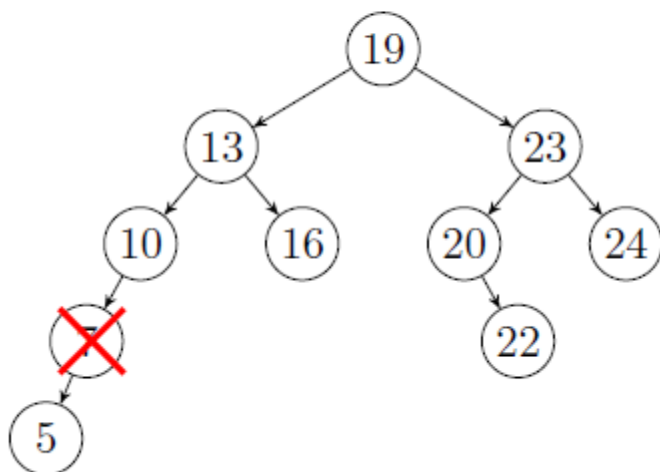
8. gyakorlat

A 11-nek nincsen gyereke: 1. eset. Simán kitörölhetjük.



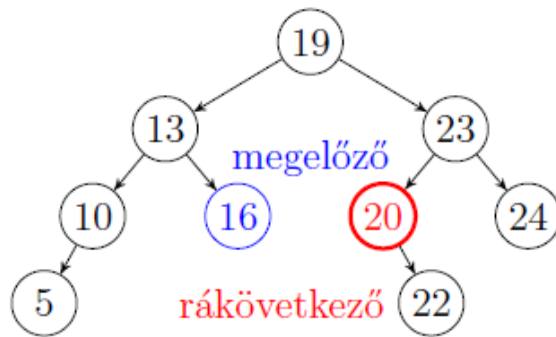
8. gyakorlat

A 7-nek egy gyereke van: 2. eset. Rakjuk a helyére az 5-öt.

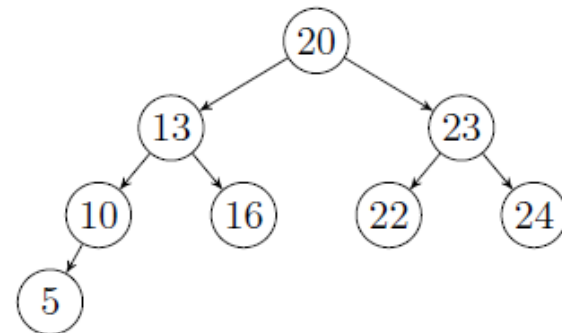


8. gyakorlat

A 19-nek két gyereke van: 3. eset. Keressük meg a rákövetkezőjét.



(a) $search(23, 19)$

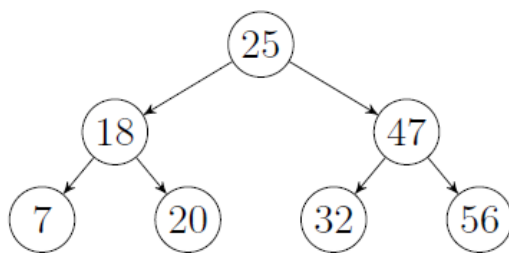


(b) 20 helyére tegyük a 22-t, 19 helyére a 20-at

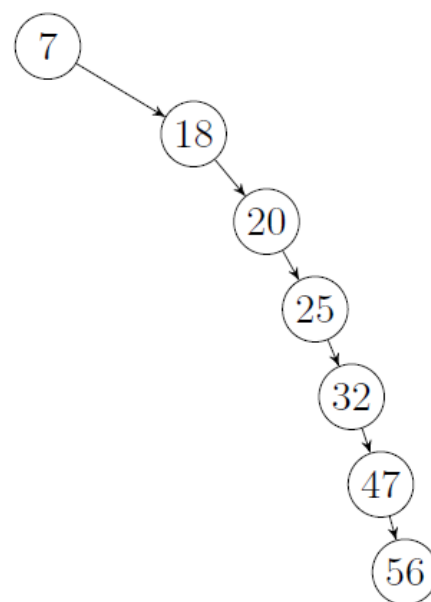
8. gyakorlat

5.Feladat: Határozzuk meg mi a minimum es maximum lehetséges magassága egy n elemű bináris keresőfának. Ehhez szúrjuk be egy fába a következő sorrendben a 25; 18; 47; 7; 20; 32; 56 elemeket és egy másikba a 7; 18; 20; 25; 32; 47; 56 sorrendben.

8. gyakorlat



(a) A 25, 18, 47, 7, 20, 32, 56 sorrend fája



(b) A 7, 18, 20, 25, 32, 47, 56 sorrend fája

8. gyakorlat

6. Írjunk olyan rekurzív függvényeljárást, amelyik kiszámítja egy kapcsolati tömb ábrázolású fa magasságát!

Házi feladat

8. gyakorlat

7. Írjunk olyan rekurzív függvényeljárást, amely adott k paraméterre kiszámítja, hogy a fának hány pontja van a k -adik szinten (vagyis azon p pontok számát, amelyekre $d(p) = k$). A fa kapcsolati tömb ábrázolású!

Házi feladat

8. gyakorlat

8. Adjunk olyan rekurzív függvényeljárást, amelyik kiszámítja a fa leveleinek számát! A fa elsőfiú-testvér ábrázolású!

Házi feladat