Algoritmusok és adatszerkezetek II.

Piros-fekete fák

Szegedi Tudományegyetem

Piros-fekete fák tulajdonságai

1. Minden csúcs színe piros vagy fekete
2. A gyökér színe fekete
3. Minden levele[1] fekete
4. A piros csúcsoknak **kizárólag** fekete színű gyerekeik vannak
5. Bármely csúcsból azonos számú fekete csúcs érintésével jutunk el bármelyik levélbe

[1] levelek alatt itt most az "őrszemeket" értjük

Piros-fekete fák tulajdonságai

1. Minden csúcs színe piros vagy fekete
2. A gyökér színe fekete
3. Minden levele[1] fekete
4. A piros csúcsoknak **kizárólag** fekete színű gyerekeik vannak
5. Bármely csúcsból azonos számú fekete csúcs érintésével jutunk el bármelyik levélbe

Tétel

Bármely n kulcsú piros-fekete fa magassága legfeljebb $2\log(n+1)$.

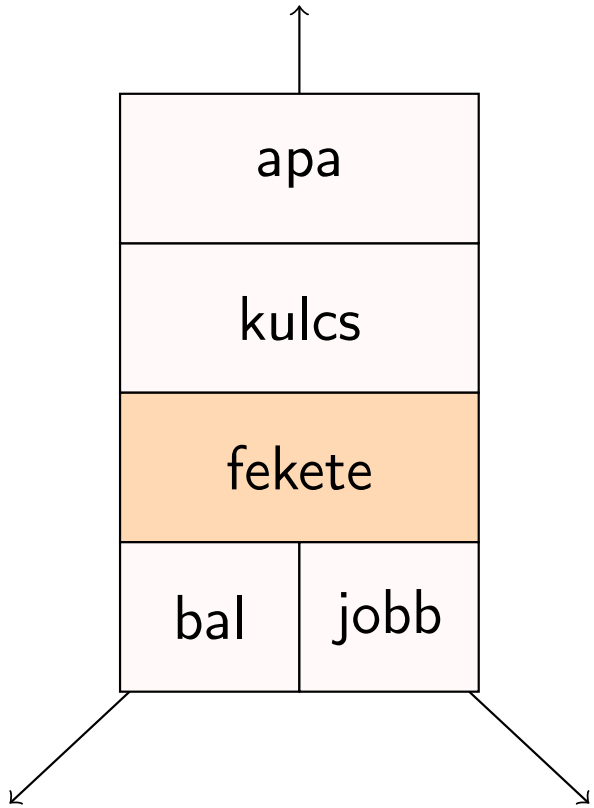
[1] levelek alatt itt most az "őrszemeket" értjük

Fekete-magasság

- $fm(x)$ jelölje az x csúcsból induló, bármely levélig vezető úton található, (x-en kívüli) fekete csúcsok számát

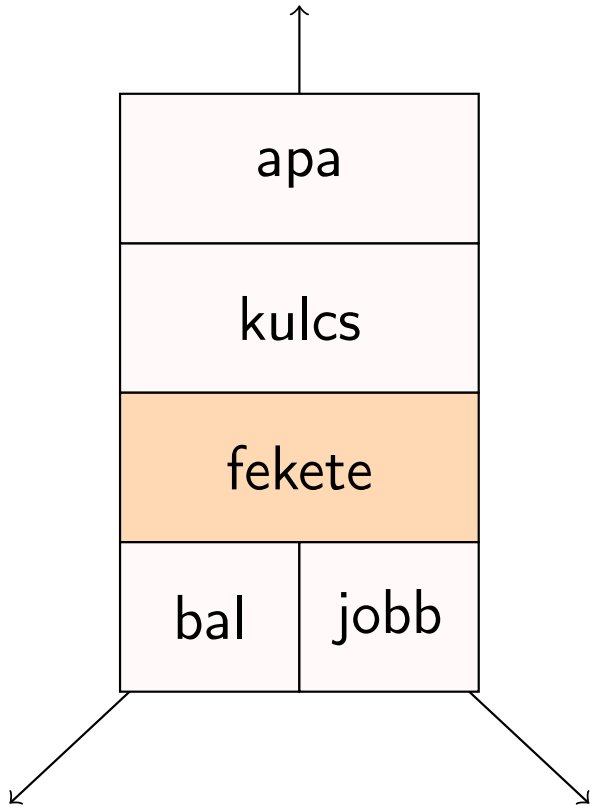
Piros-fekete fa implementációja

```
class Node {
    Object kulcs;
    boolean fekete;
    Node *apa;
    Node *bal;
    Node *jobb;
}
```



Piros-fekete fa implementációja

```
class Node {
    Object kulcs;
    boolean fekete;
    Node *apa;
    Node *bal;
    Node *jobb;
}
```



Megjegyzés

Az eddigi kiegészítő információk közül a legolcsóbb (csupán 1 bit)

Összegzés

- A bináris keresőfák műveletei $O(h)$ idejűek
- Legrosszabb esetben azonban n is lehet a fák magassága ($\Theta(\log n)$ helyett)
- Kiegyensúlyozott keresőfák használatával garantálható, hogy a keresőfa kiegyensúlyozottsága sose romoljon el "túlságosan"