ADATSZERKEZETEK ÉS ALGORITMUSOK

Keresések hatékonysága

Keresések

- Sok adat esetén milyen adatszerkezetben lehet hatékonyan
  - keresni,
  - módosítani,
  - beszúrni és
  - törölni?
- A gyakorlat azt mutatja, hogy fákban és táblázatokban

Keresések

- Szekvenciális keresések
  - a keresési idő n-nel arányos: $\mathcal{O}(n)$
  - rendezetlen tömbök
  - láncolt listák
- Bináris keresés
  - rendezett tömbök
  - a keresési idő $\log_2 n$-nel arányos: $\mathcal{O}(\log_2 n)$

Keresések

- Rendezett tömb létrehozása
  - elem hozzáadása megfelelő helyre
    - megkeresem a pozícióját: $c_1 \log_2 n$
    - eltolom: $c_2 n$
    - összesen: $c_1 \log_2 n + \boxed{c_2 n}$
    - vagyis: $c_2 n$
  - Minden elem hozzáadása a rendezett tömbhöz: $\mathcal{O}(n)$
- Lehet-e a rendezett tömböt olcsóbb beszúrásokkal karbantartani?

dominháns

Keresések

- Szótár (dictionary) egy adatszerkezet, ha értelmezve vannak a következő műveletek:
  - Keres
  - Beszúr
  - Töröl
  - (Tól-ig)

Keresések

- Prioritásos sor egy adatszerkezet, ha az előzőeken kívül értelmezve vannak a következők is
  - Minimum
  - Maximum
  - Előző
  - Rákövetkező
- Ezzel lehet rendezni

Keresések

- Adatok a struktúrában
  - kulcs + mezők (rekordok)
- Lehetőségek
  - Kulcsegyezőség
    - I. minden kulcs különböző
    - II. lehetnek azonos kulcsok
  - Adatstruktúra
    - A. rekordok: $(k, m_1, m_2, \dots, m_n)$
    - B. csak a kulcsokat nézzük: k
  - Mi most az I. és B.-t választjuk.

Bináris keresőfák

- Mennyire hatékony a bináris keresőfa?
  - Minden művelet egy útvonal bejárását jelenti a fában
  - A h magasságú fákban $\mathcal{O}(h)$ idő alatt hajtódnak végre
  - Ahogy elemeket beszúrunk és törlünk, változik a fa magassága és ezzel a műveletek ideje is
    - Itt ugyanis nem tudom a fa átlagos magasságát
    - Ha csak beszúrások vesszük a felépítés során, akkor könnyebben elemezhető

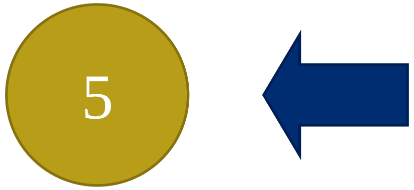
Bináris keresőfák

- Legyen adva n különböző kulcs, ebből bináris keresőfát építünk.
  - Ha itt minden sorrend egyformán valószínű, akkor a kapott fát véletlen építésű bináris keresőfának nevezzük.
- Bizonyítható, hogy egy n kulcsból **véletlen módon** épített bináris keresőfa átlagos magassága $\mathcal{O}(\log_2 n)$.

Bináris keresőfák

- Tegyük fel, hogy a véletlen sorrendű $1,2,\ldots n$ adatokból építjük fel a t keresőfát.
- Mennyi az átlagos csúcsmagasság?
- Ennek megválaszolása megadja a következő kérdésre is a megoldást:
  - Hány összehasonlítással lehet felépíteni a t keresőfát átlagosan?
- A meghatározáshoz vegyünk egy keresőfát

Bináris keresőfák



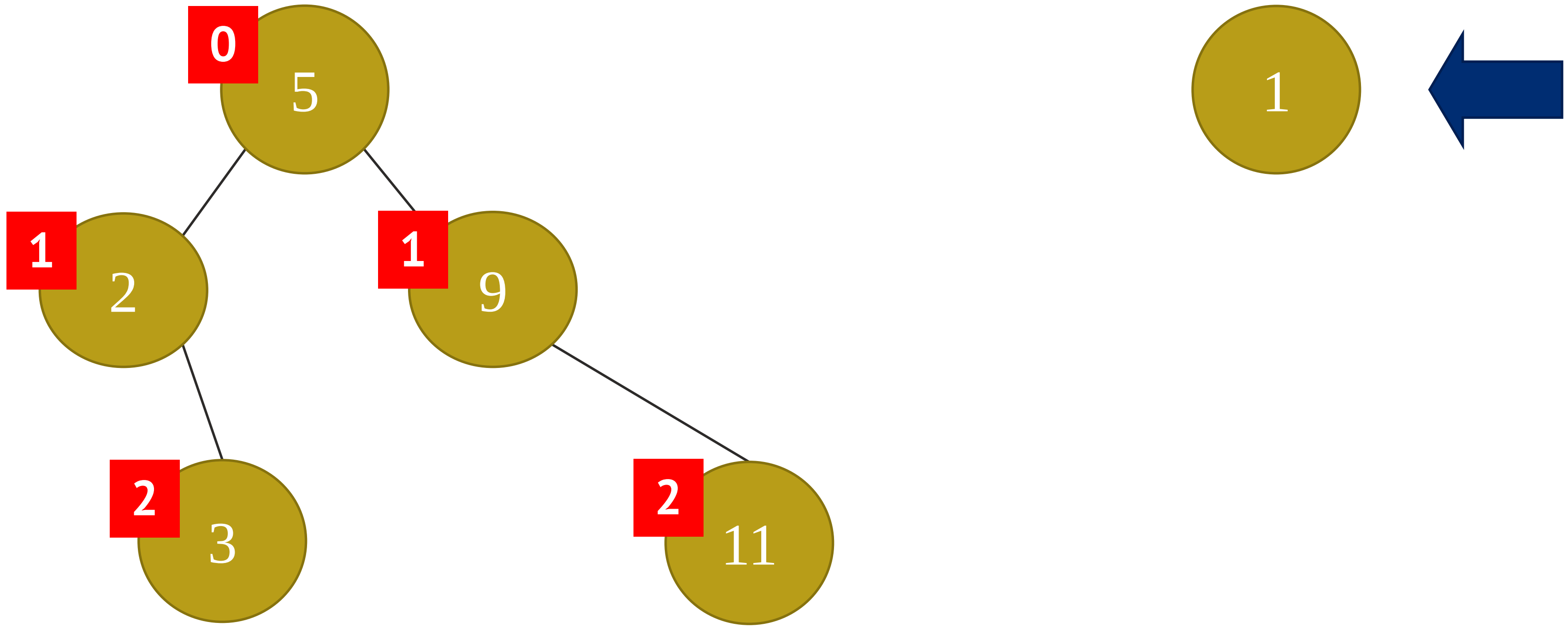
Bináris keresőfák

Bináris keresőfák

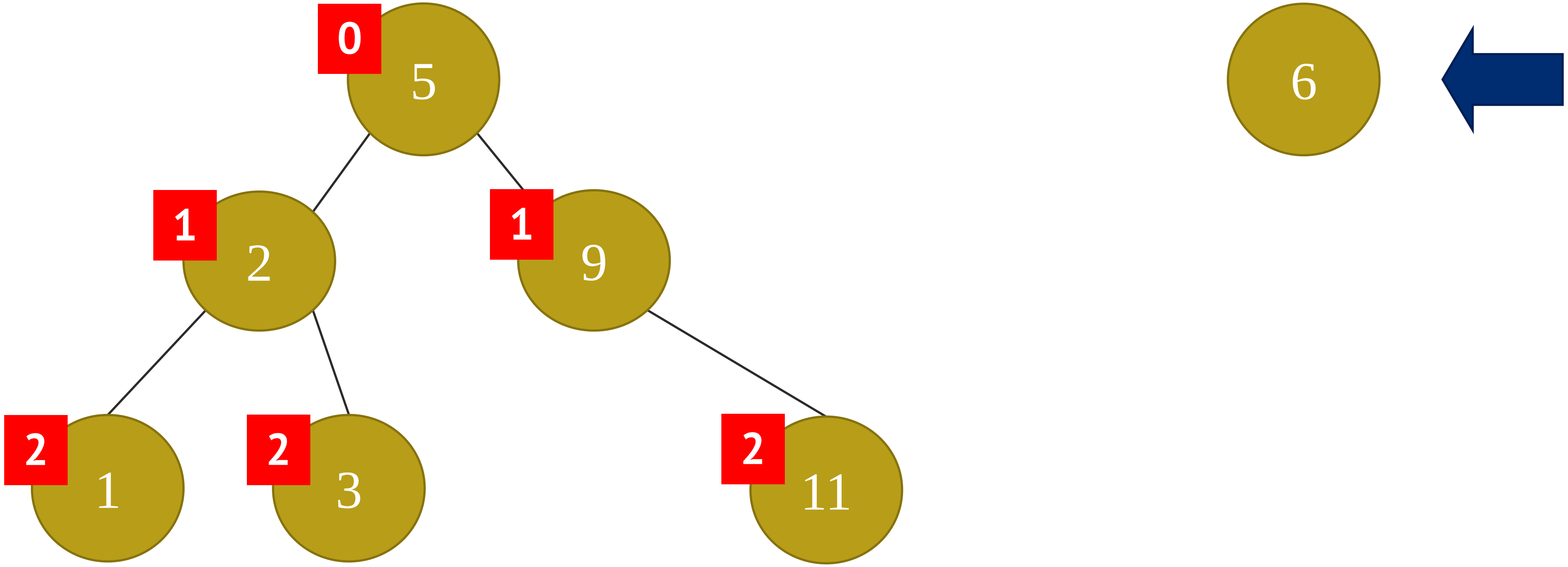
Bináris keresőfák

Bináris keresőfák

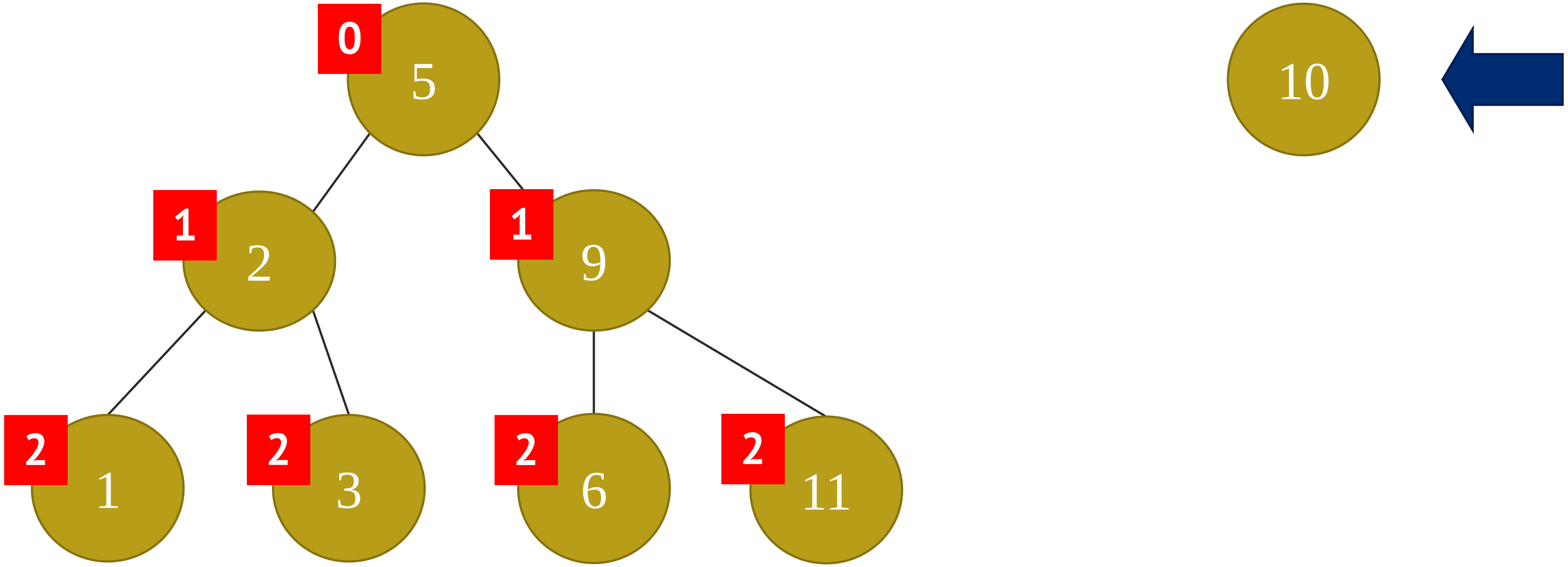
Bináris keresőfák



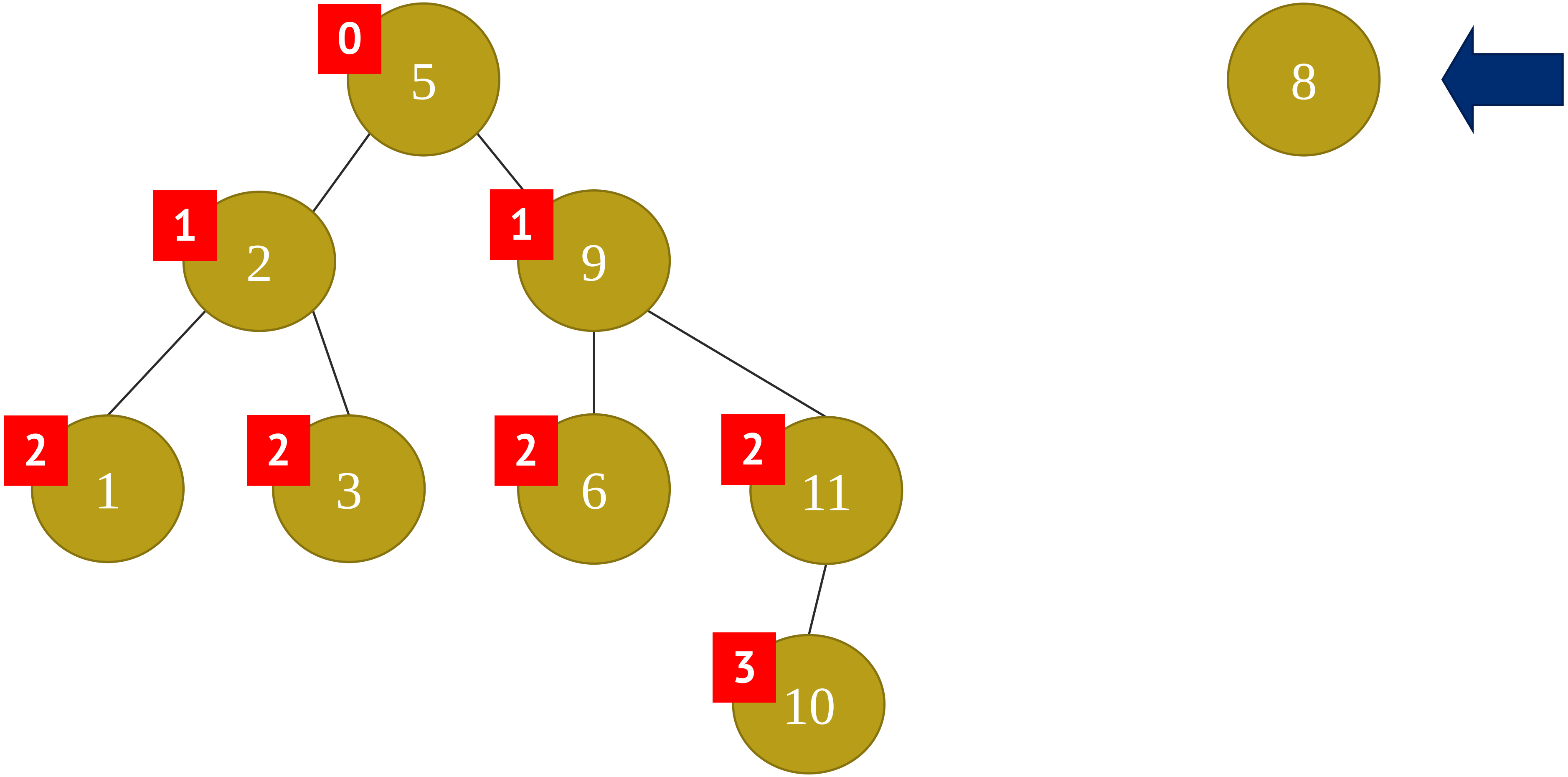
Bináris keresőfák



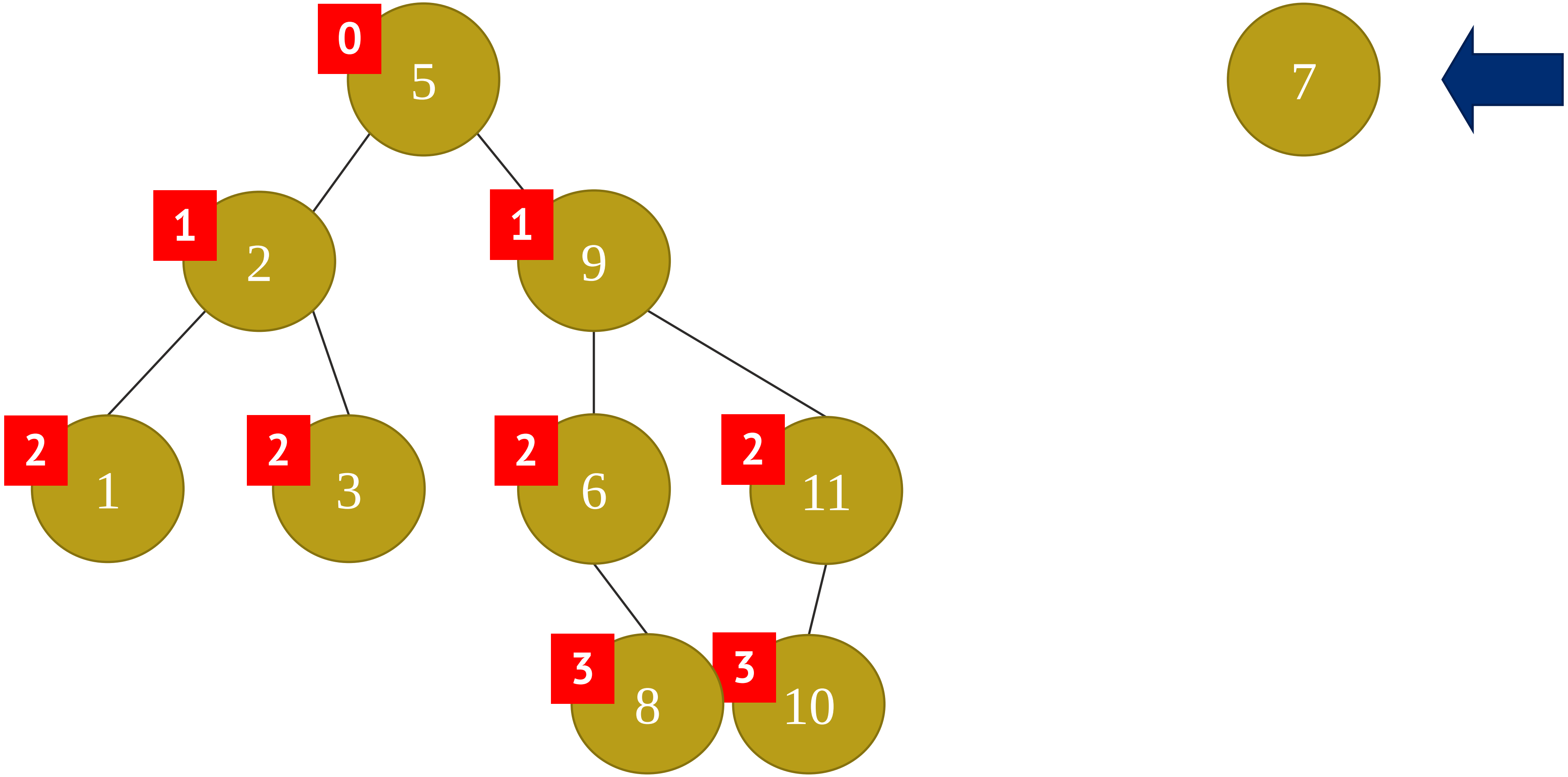
Bináris keresőfák



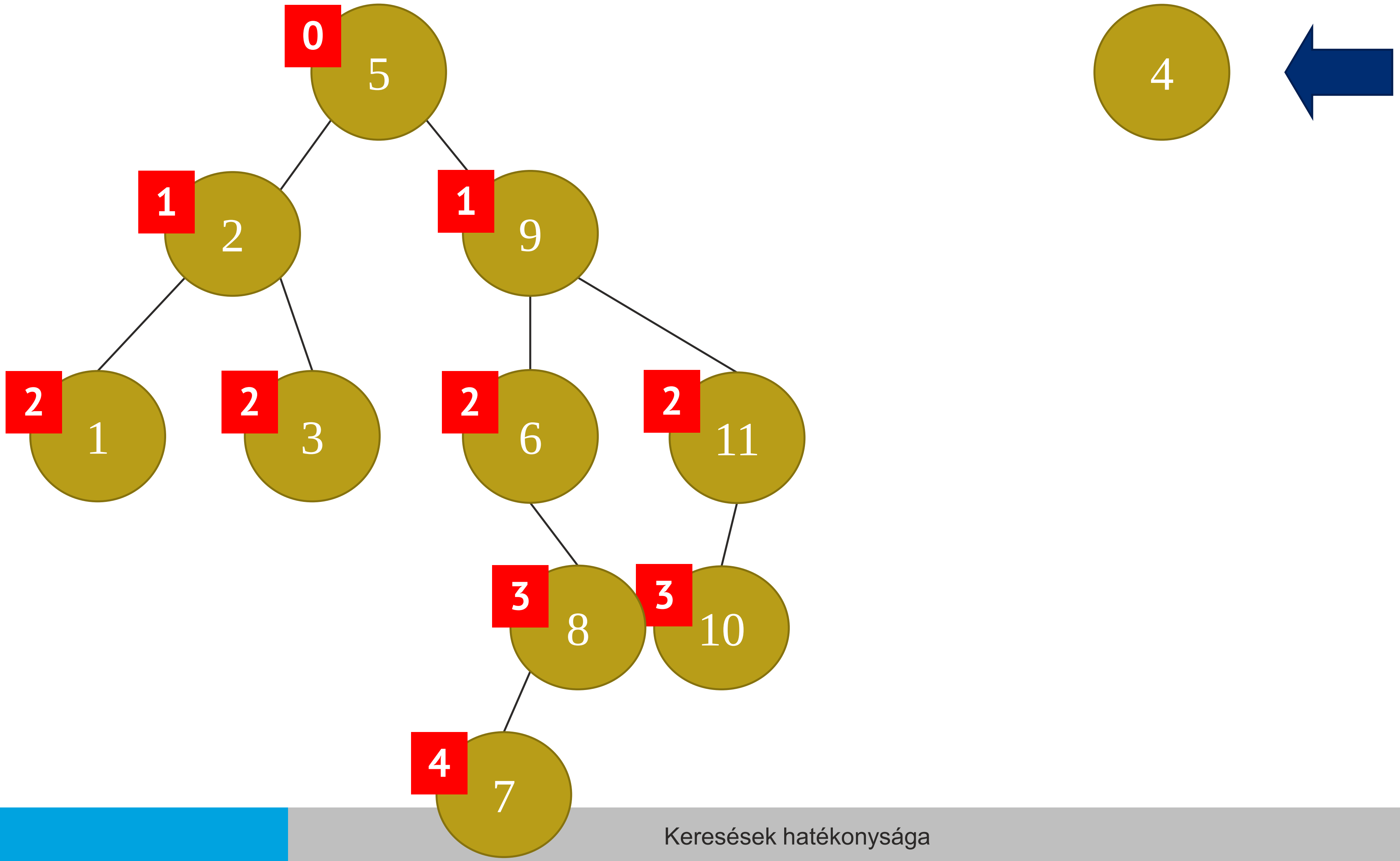
Bináris keresőfák



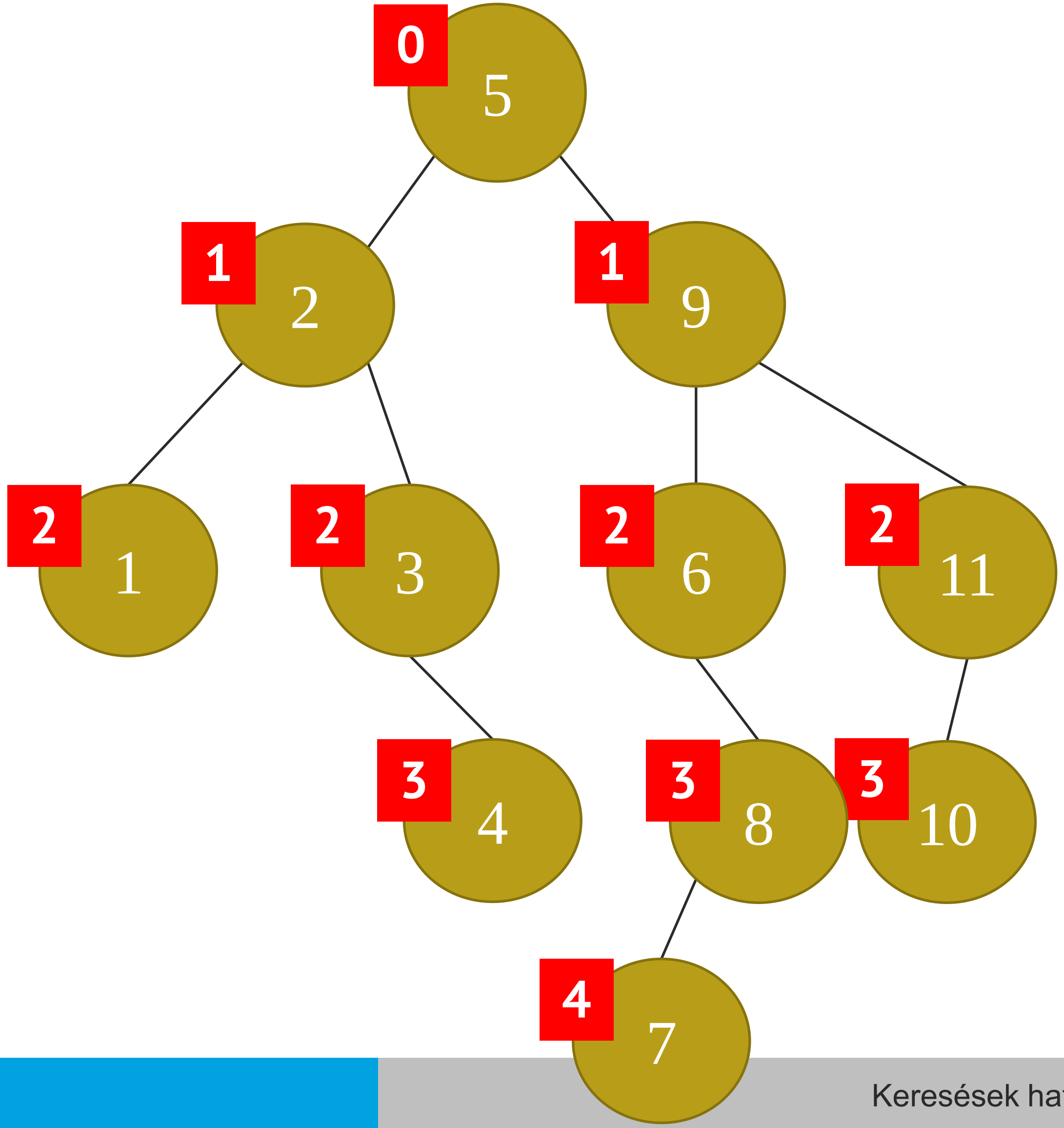
Bináris keresőfák



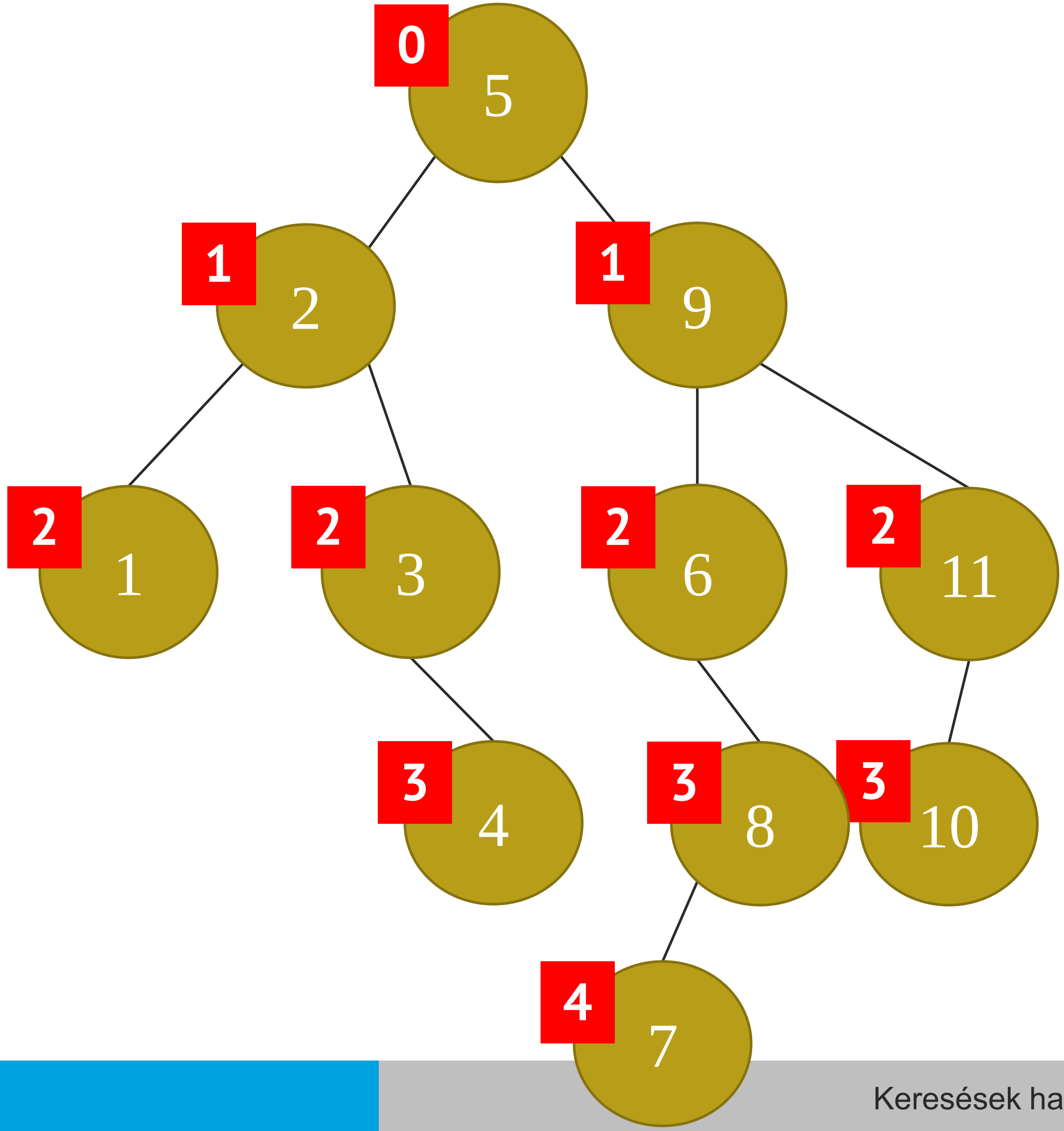
Bináris keresőfák



Bináris keresőfák



Bináris keresőfák



$p = 5,2,9,3,11,1,6,10,8,7,4$

esetén

$$Ö(p) = (1+1) + (2+2+2+2) + (3+3+3) + 4 = 23$$

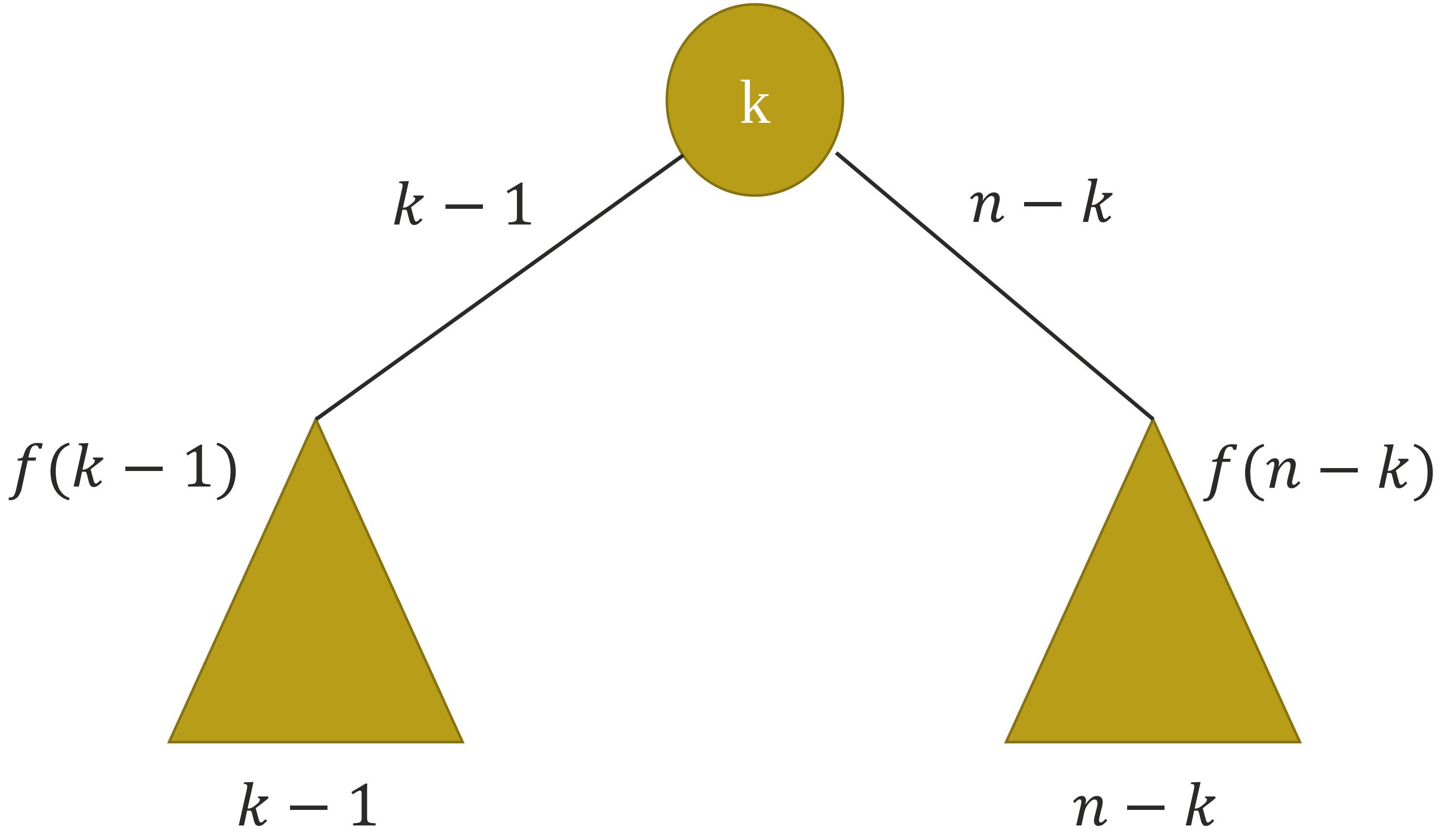
Bináris keresőfák

- Határozzuk meg ennek az átlagát!
- Először meghatározzuk a csúcsmagasság összeget.
- Jelölés:
  - $f(n)$: n adatból hány összehasonlítással lehet keresőfát építeni
  - $f(n|k)$: először a k érték jön (k az input sorozat első eleme)
  - Tegyük fel, hogy minden sorozat egyforma valószínűségű

$$f(n) = \frac{1}{n} \sum_{k=1}^{n} f(n|k)$$

Bináris keresőfák

- Összehasonlítások



Bináris keresőfák

$$f(n) = \frac{1}{n}\sum_{k=1}^{n}\left(k - 1 + f(k-1) + (n-k) + f(n-k)\right)$$

$$\Downarrow$$

$$f(0) = 0$$

$$f(n) = (n-1) + \frac{2}{n}\sum_{k=1}^{n-1} f(k)$$

Belátható, hogy $f(n) < 2n * \ln n \approx 1{,}39n \log_2$n

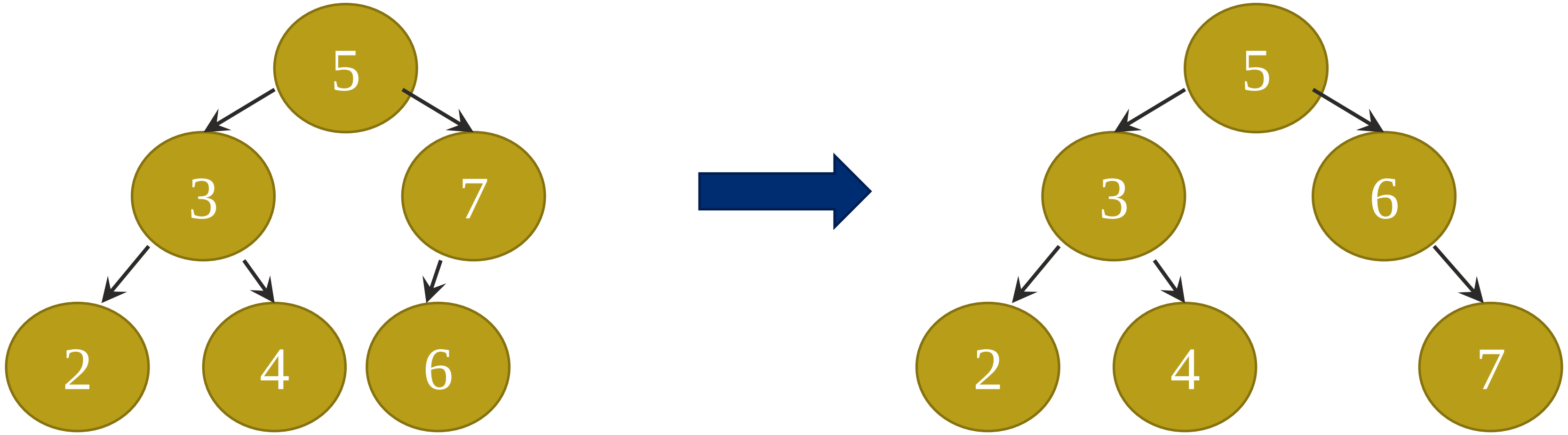
Tehát a fa csúcsmagasság összege $\approx 1{,}39n \log_2 n$

Bináris keresőfák

- Tehát a fa csúcsmagasság összege $\approx 1{,}39n \log_2 n$
- Ebből következik, hogy a véletlen kulcssorozatból készítette bináris keresési fa várható csúcsmagasság-átlaga $\approx 1{,}39 \log_2 n$
  - Ez jó eredmény, mivel $1{,}39 \log_2 n = \mathcal{O}(\log_2 n)$.
    - Rosszabb, mint az optimális $h = \log_2 n$
    - Jobb, mint a $h = n$ szélsőséges eset
- Fontos, hogy ez csak akkor érvényes, ha a kulcsok véletlen sorrendben érkeznek, ami a valóságban általában nem teljesül.

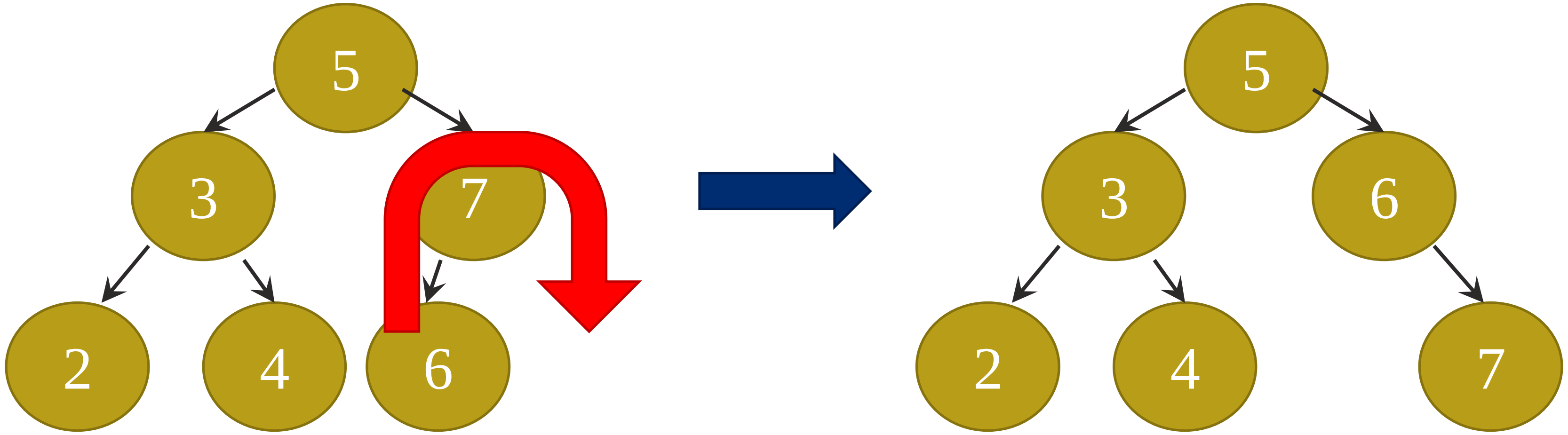
Bináris keresőfák

- A sorrend megőrzése fontos
  - ez a transzformáció megőrzi a keresőfa tulajdonságot



Bináris keresőfák

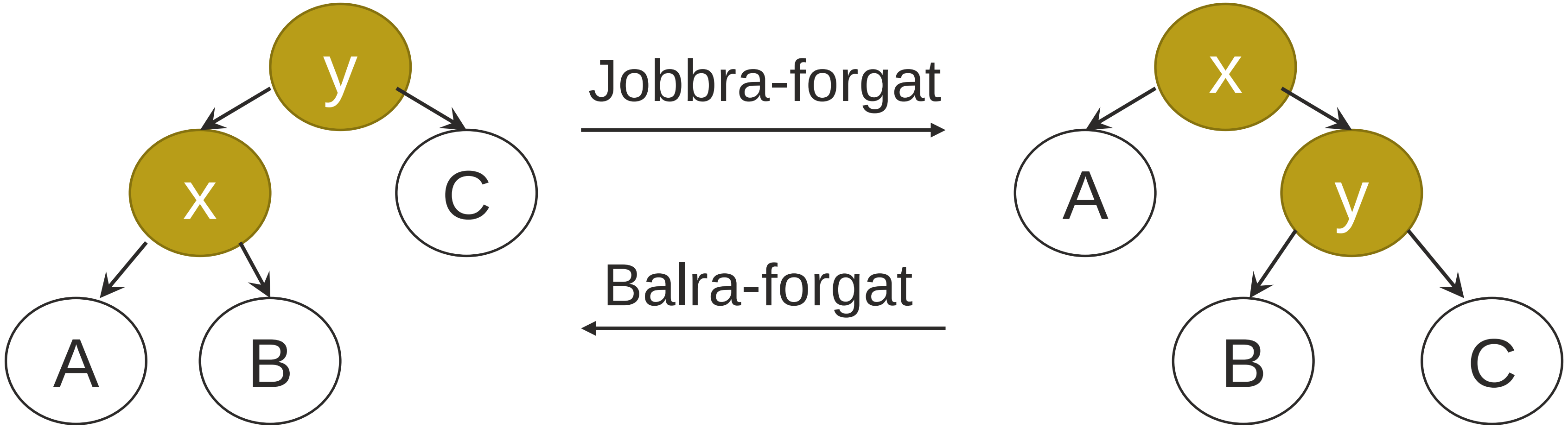
- A sorrend megőrzése fontos
  - ez a transzformáció megőrzi a keresőfa tulajdonságot



  - Forgatást (jobbra) hajtottunk végre a 7 és 6 csúcsok körül

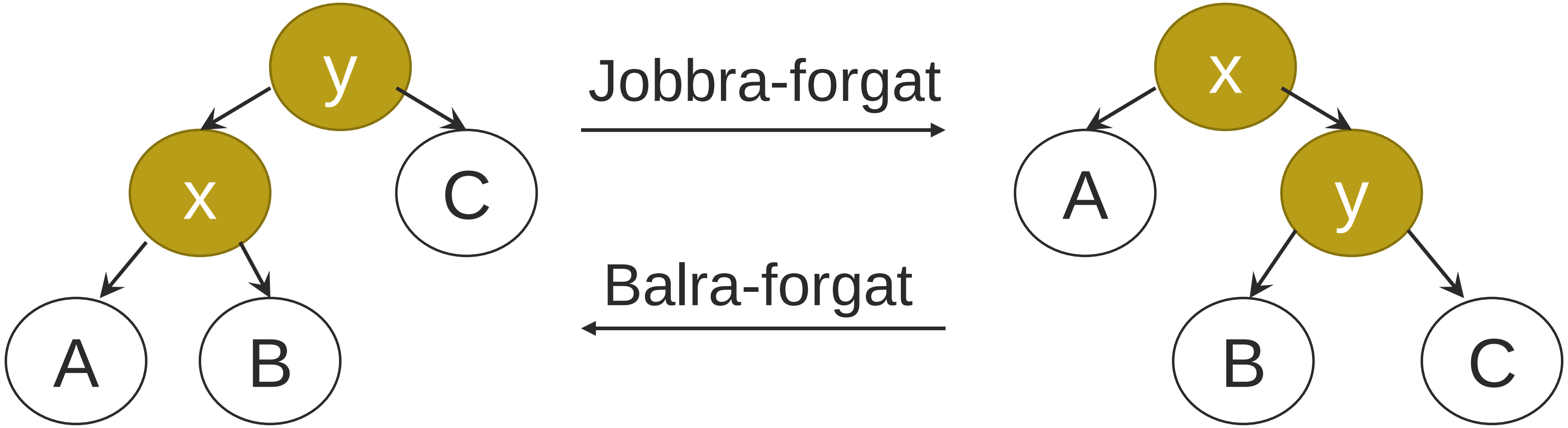
Bináris keresőfák

- A forgatások lehetnek bal- vagy jobb-forgatások
- Mindkét fára az inorder bejárás
  - A x B y C



Bináris keresőfák

- A forgatások lehetnek bal- vagy jobb-forgatások
- Mindkét fára az inorder bejárás
  - A x B y C



- Az x-y viszony megváltozásán túl a B részfa helyzete is változik
  - A x jobbgyerekéből átkerül az y balgyerekébe

AVL fa

Következő téma