

# Adatszerkezetek

## 09. Kupacok

Vekov Géza

2023. április 26.



Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

Kupac

# Kupac (*heap*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Adatok

- Olyan elemeket kell eltárolni, amelyek egy *kulccsal* rendelkeznek.
- **Fontos:** a kulcsokat össze lehet hasonlítani egymással (<, >, = operátorokkal).

# Kupac (*heap*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Adatok

- Olyan elemeket kell eltárolni, amelyek egy *kulccsal* rendelkeznek.
- **Fontos:** a kulcsokat össze lehet hasonlítani egymással ( $<$ ,  $>$ ,  $=$  operátorokkal).

## Cél

- Határozzuk meg gyorsan a minimális / maximális kulcsú elemet!

# Kupac (*heap*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Adatok

- Olyan elemeket kell eltárolni, amelyek egy *kulccsal* rendelkeznek.
- **Fontos:** a kulcsokat össze lehet hasonlítani egymással ( $<$ ,  $>$ ,  $=$  operátorokkal).

## Cél

- Határozzuk meg gyorsan a minimális / maximális kulcsú elemet!

## Műveletek

- Beszúrás
- Törlés
- Minimum VAGY maximum keresés

# Kupac (*heap*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Adatok

- Olyan elemeket kell eltárolni, amelyek egy *kulccsal* rendelkeznek.
- **Fontos:** a kulcsokat össze lehet hasonlítani egymással ( $<$ ,  $>$ ,  $=$  operátorokkal).

## Cél

- Határozzuk meg gyorsan a minimális / maximális kulcsú elemet!

## Műveletek

- Beszúrás
- Törlés
- Minimum VAGY maximum keresés -  $O(1)$

## Kupac szerkezet

### Majdnem teljes bináris fa.

- Minden csomópontnak max. két gyereke van
- Minimális magasságú
- Ha a legalsó szint nincs teljesen kitöltve, akkor azon a szinten a csomópontok balról jobbra kerülnek feltöltésre

## Kupac szerkezet

### **Majdnem teljes bináris fa.**

- Minden csomópontnak max. két gyereke van
- Minimális magasságú
- Ha a legalsó szint nincs teljesen kitöltve, akkor azon a szinten a csomópontok balról jobbra kerülnek feltöltésre

## Kupactulajdonság

### **Minden szülő nagyobb / kisebb, mint a gyermekei.**

- Bármely adatelem kulcsa kisebb vagy egyenlő az adatelem szülőjének a kulcsával (maximum-kupac).
- A gyökérelem mindig a legnagyobb elem.
- A csomópontok sorrendje nincs meghatározva, ezért ezek tetszőlegesen felcserélhetőek, ha nem sértik a kupactulajdonságot.



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupac AAT

A továbbiakban:

**kupac**  $\Rightarrow$  bináris kupac

# Kupac

Adatszerkezetek

Vekov Géza

Kupac

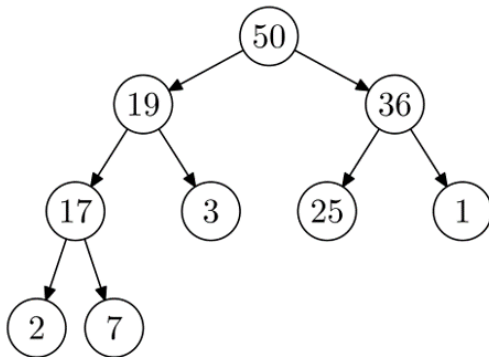
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

Példa 1.



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

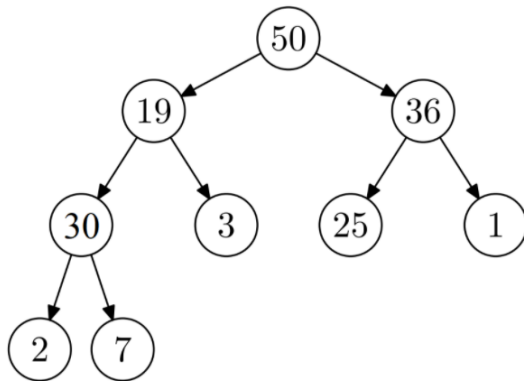
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

Példa 2.



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

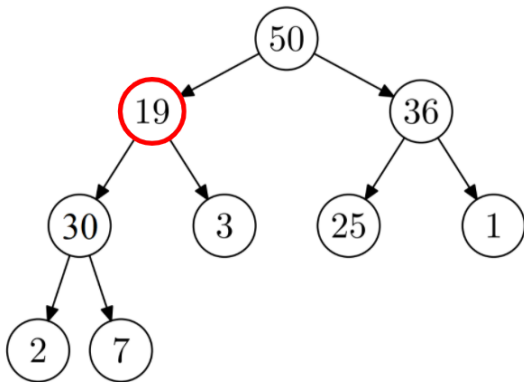
Műveletek

Alkalmazások

Elsőbbségi sor

## Példa 2.

- **NEM kupac:** sérül a kupactulajdonság



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

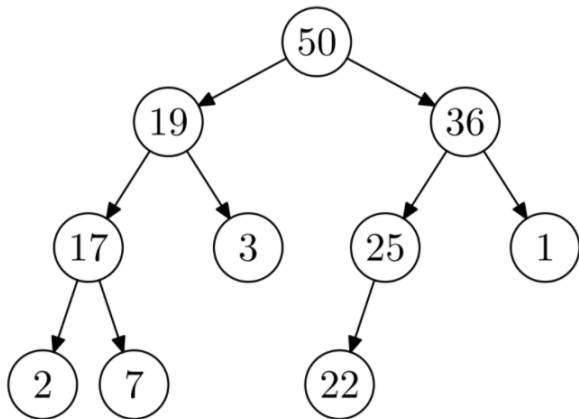
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

Példa 3.



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

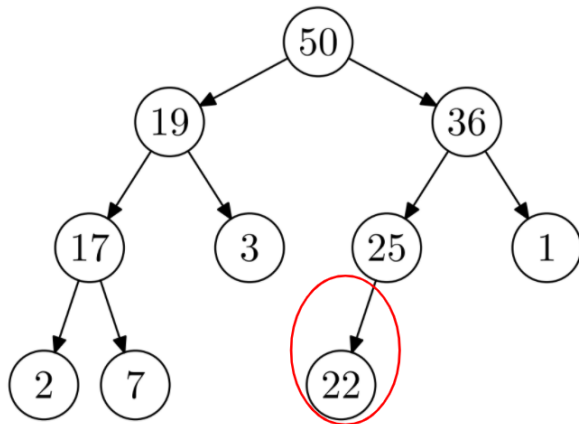
Műveletek

Alkalmazások

Elsőbbségi sor

## Példa 3.

- **NEM kupac:** sérül a kupac szerkezet



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

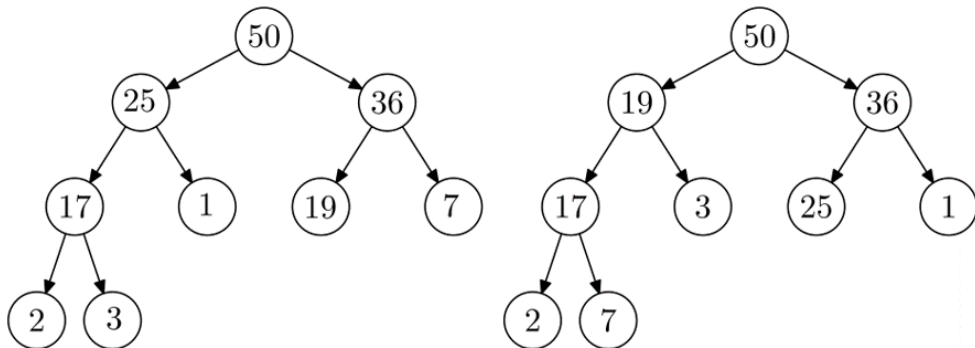
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Példa 4.



# Kupac

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

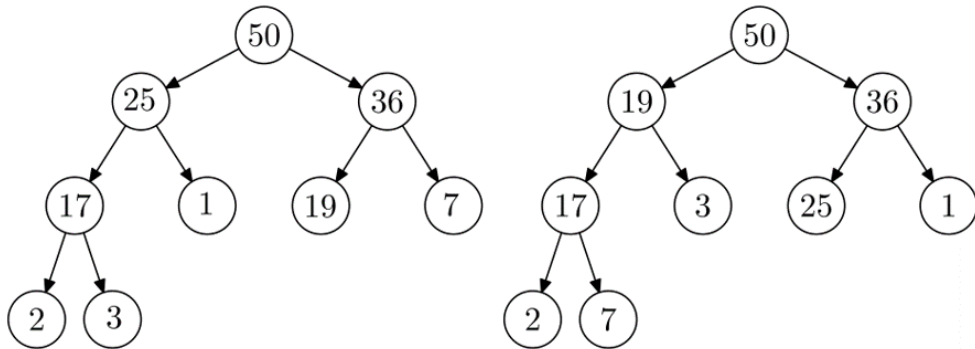
Műveletek

Alkalmazások

Elsőbbségi sor

## Példa 4.

- a csomópontok sorrendje tetszőleges, ha teljesül a kupac tulajdonság





# Kupactípusok

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Bináris maximumkupac

- minden elemnek legfeljebb két gyereke van
- *kupactulajdonság*: szülő *nagyobb* a gyerekeinél

# Kupactípusok

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Bináris maximumkupac

- minden elemnek legfeljebb két gyereke van
- *kupactulajdonság*: szülő *nagyobb* a gyerekeinél

## Bináris minimumkupac

- minden elemnek legfeljebb két gyereke van
- *kupactulajdonság*: szülő *kisebb* a gyerekeinél

# Kupactípusok

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Bináris maximumkupac

- minden elemnek legfeljebb két gyereke van
- *kupactulajdonság*: szülő *nagyobb* a gyerekeinél

## Bináris minimumkupac

- minden elemnek legfeljebb két gyereke van
- *kupactulajdonság*: szülő *kisebb* a gyerekeinél

## Egyéb kupactípusok

- binomiális kupac, Fibonacci-kupac, intervallumkupac, párosító kupac, stb.

# Kupac: ábrázolás

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

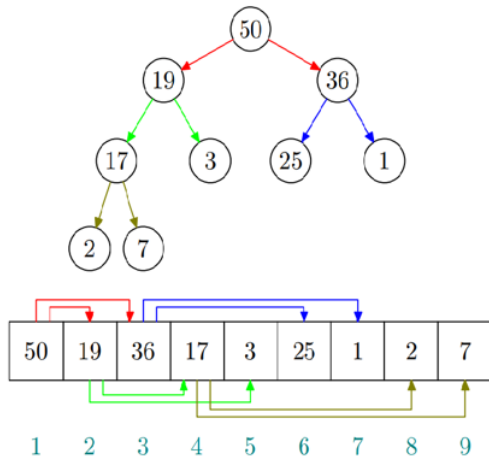
Műveletek

Alkalmazások

Elsőbbségi sor

## Ábrázolás

- Szerkezete miatt a kupacot leggyakrabban egy *egydimenziós tömbben* tároljuk



# Kupac: ábrázolás

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

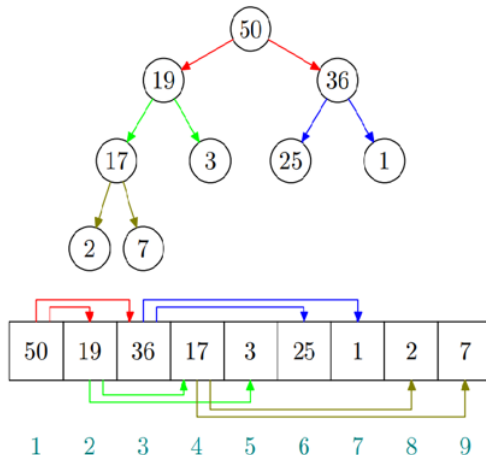
Alkalmazások

Elsőbbségi sor

## Ábrázolás

- Szerkezete miatt a kupacot leggyakrabban egy *egydimenziós tömbben* tároljuk
- Nincs szükség mutatókra
- bármely adatelem szülőjének és gyerekének az indexe egyszerű számtani műveletekkel meghatározható
  - szülő(i):  $[i/2]^*$
  - gyerek(i):  $2 \cdot i, 2 \cdot i + 1^*$

\* 0-ás indexelésnél kis változtatással ugyanúgy meg lehet kapni az indexeket.



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Beszúrás

- Beszúrjuk az elemet az utolsó helyre
- Visszaállítjuk a kupactulajdonságot
  - A szülőre ellenőrizzük a tulajdonság teljesülését
  - Ha nem teljesül, akkor szintenként "felemeljük" az új elemet, és újraellenőrizzük, amíg visszaáll nem áll a tulajdonság

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Beszúrás

- Beszúrjuk az elemet az utolsó helyre -  $O(1)$
- Visszaállítjuk a kupactulajdonságot -  $O(\log n)$ 
  - A szülőre ellenőrizzük a tulajdonság teljesülését
  - Ha nem teljesül, akkor szintenként "felemeljük" az új elemet, és újraellenőrizzük, amíg visszaáll nem áll a tulajdonság

## Futási idő: összesítve

- Futási idő:  $O(\log n)$

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

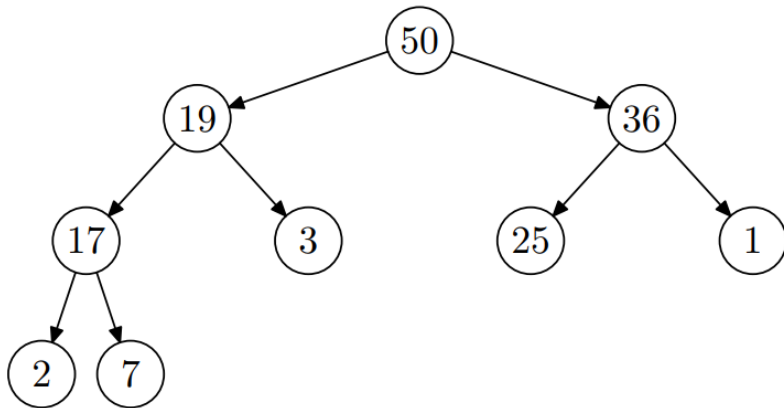
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 2-es**





# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

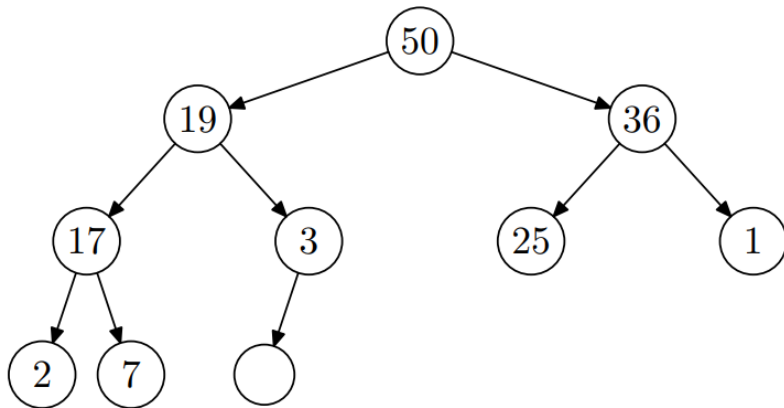
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 2-es**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

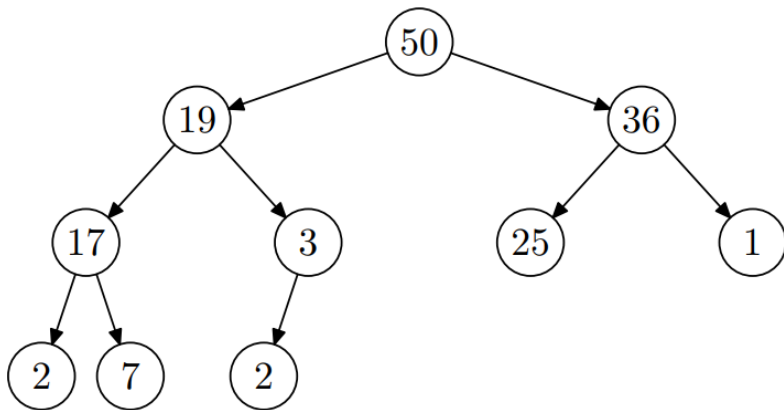
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 2-es**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

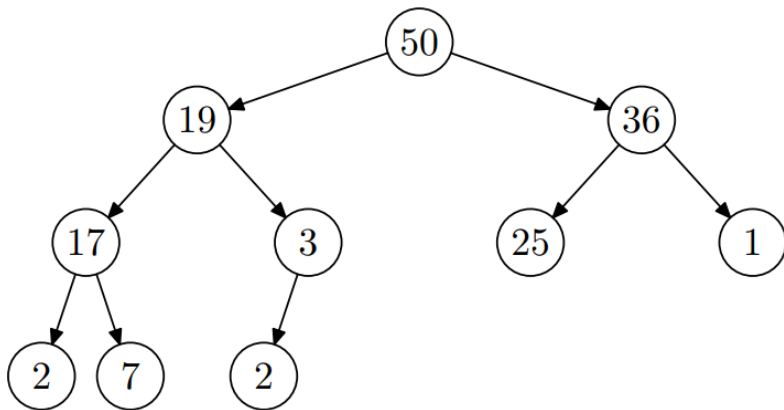
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás:** 30-as



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

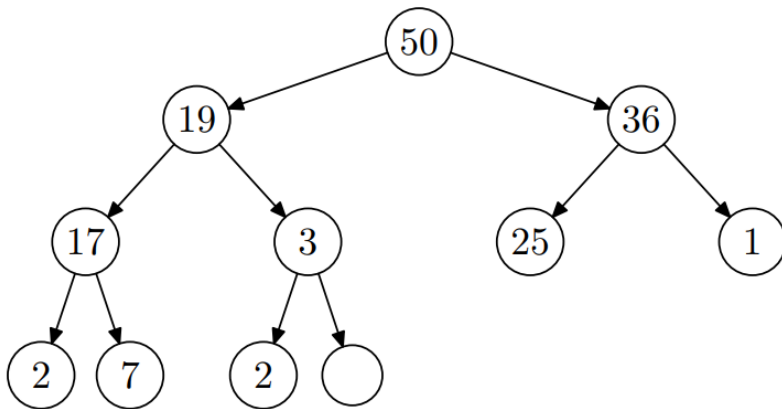
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 30-as**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

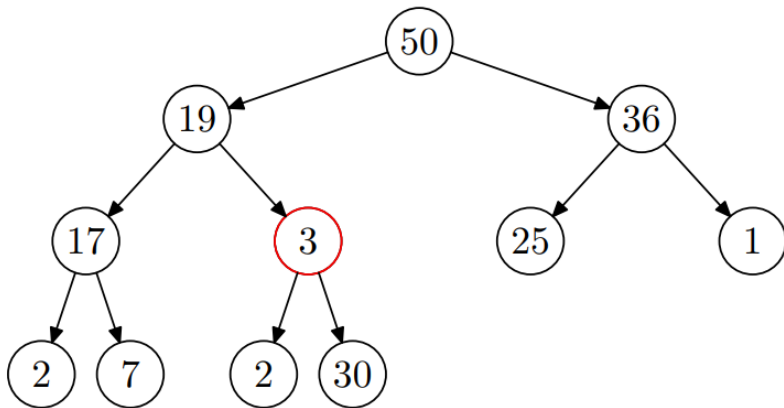
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 30-as**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

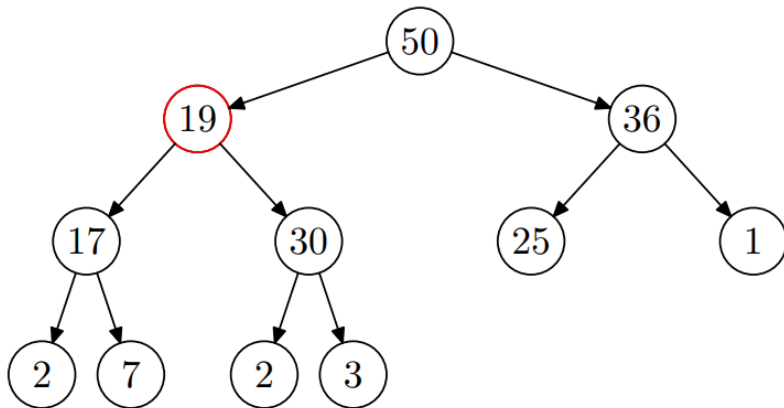
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 30-as**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

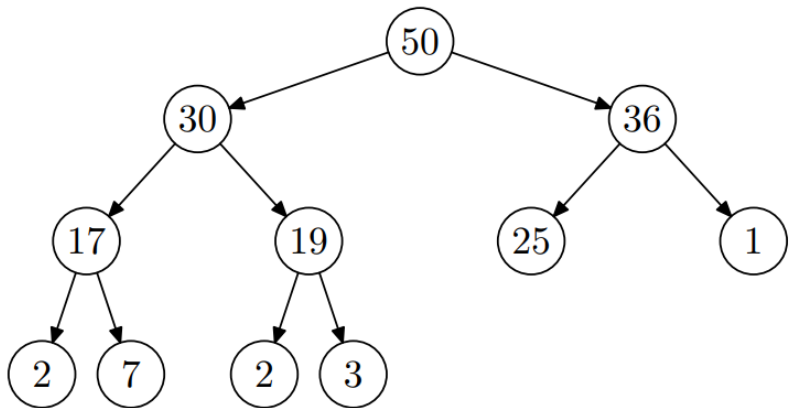
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 30-as**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

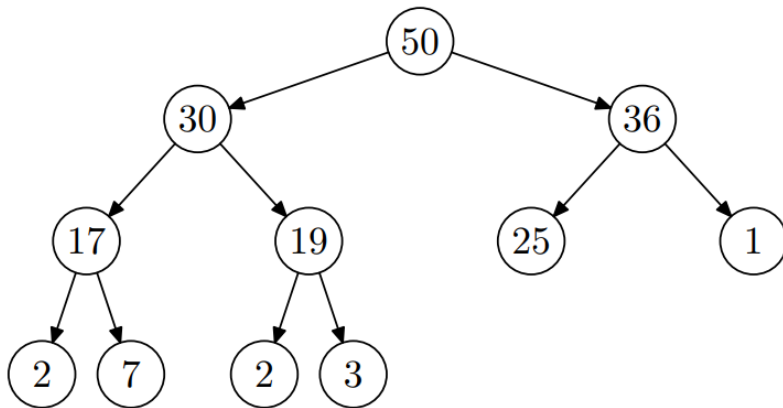
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás:** 60-as





# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

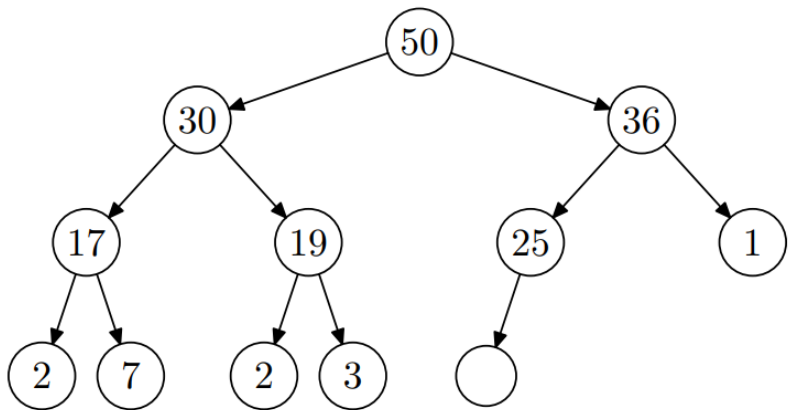
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás:** 60-as



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

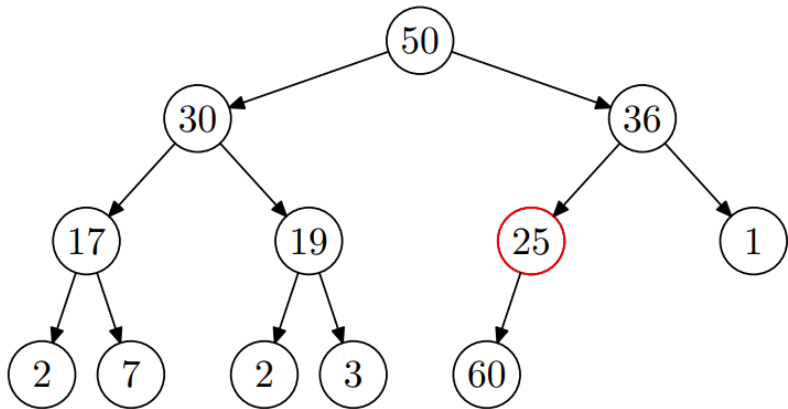
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás: 60-as**



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

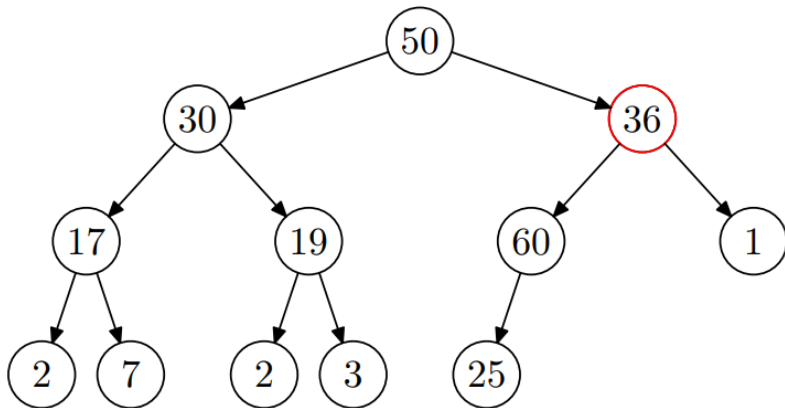
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás:** 60-as



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

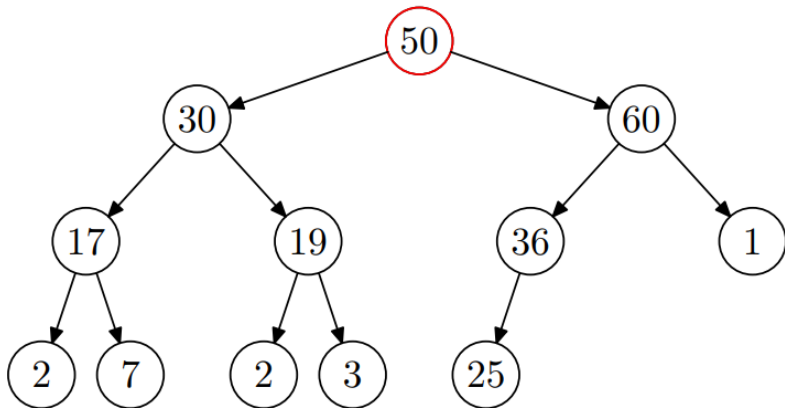
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás:** 60-as



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

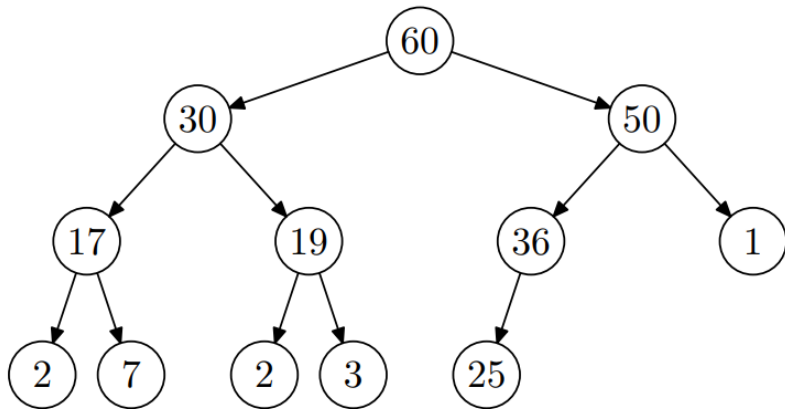
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Beszúrás:** 60-as



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Beszúrás

- Beszúrjuk az elemet az utolsó helyre -  $O(1)$
- Visszaállítjuk a kupactulajdonságot -  $O(\log n)$ 
  - A szülőre ellenőrizzük a tulajdonság teljesülését
  - Ha nem teljesül, akkor szintenként "felemeljük" az új elemet, és újraellenőrizzük, amíg visszaáll nem áll a tulajdonság

## Futási idő: összesítve

- Futási idő:  $O(\log n)$

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kulcs növelése / csökkentése ( $i$ )

A beszúráshoz hasonlóan:

- Módosítjuk az  $i$ . kulcs értékét -  $O(1)$
- Visszaállítjuk a kupactulajdonságot -  $O(\log n)$ 
  - A szülőre ellenőrizzük a tulajdonság teljesülését
  - Ha nem teljesül, akkor szintenként "felemeljük" a módosított értékű elemet, és újraellenőrizzük, amíg visszaáll nem áll a tulajdonság

## Megjegyzés

- Gyakorlatilag a beszúrást úgy is lehet értelmezni, mint  $-\infty$  értékű elem hozzáfűzése, majd *Kulcs növelése* ( $n$ )

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

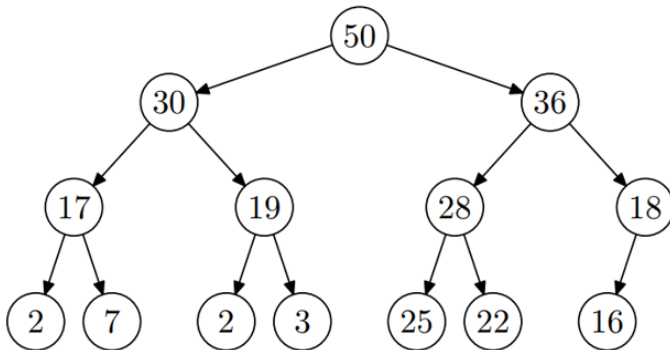
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése





# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

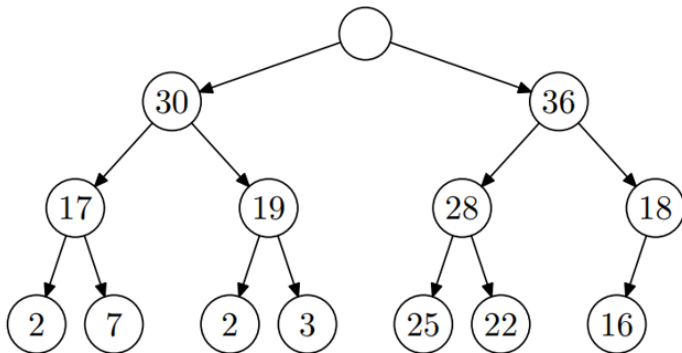
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

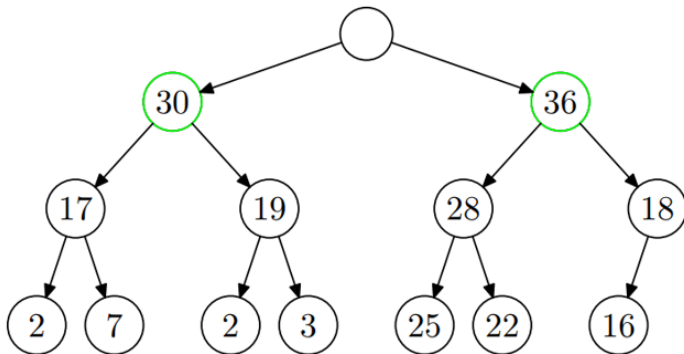
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

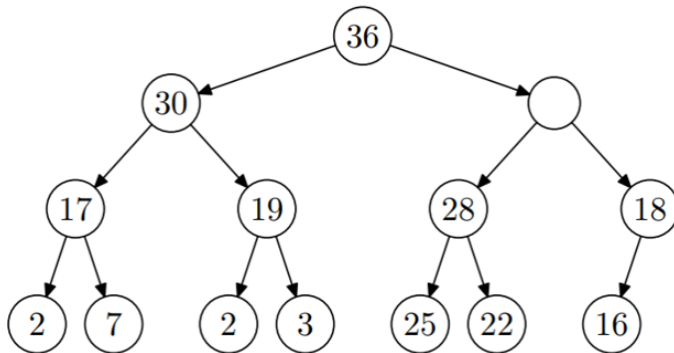
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

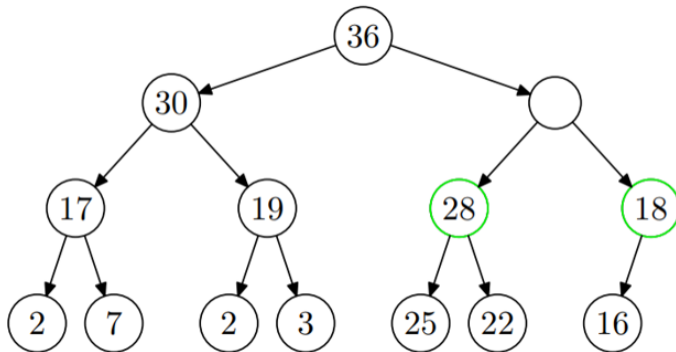
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

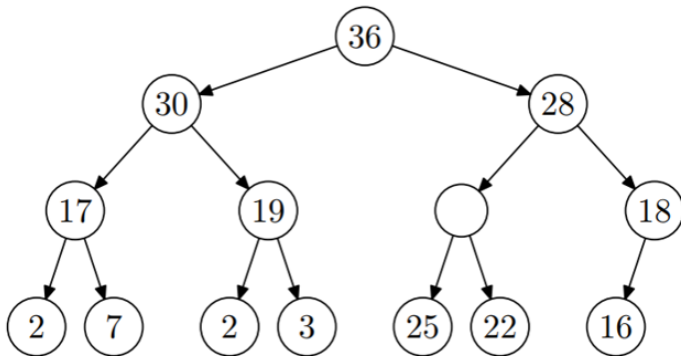
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

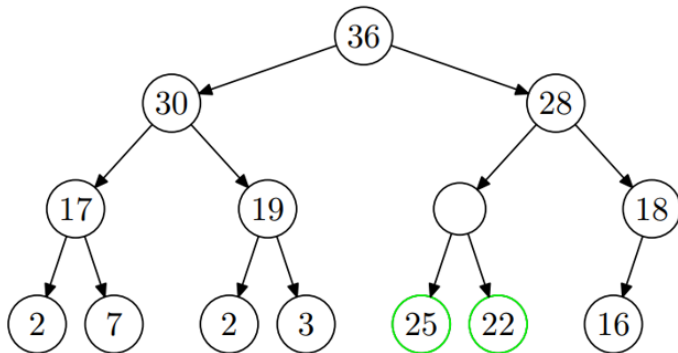
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

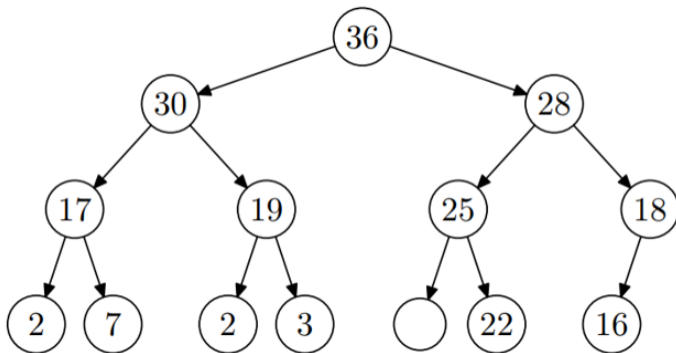
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

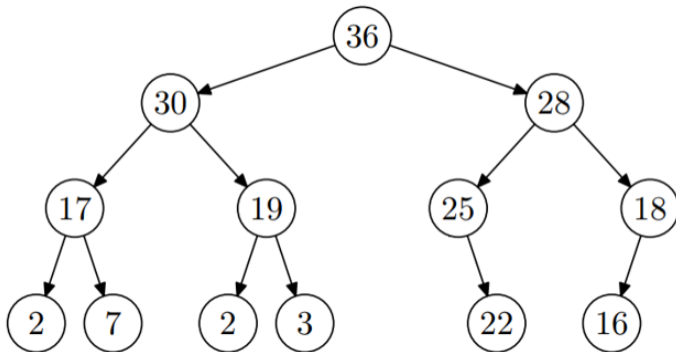
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése





# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

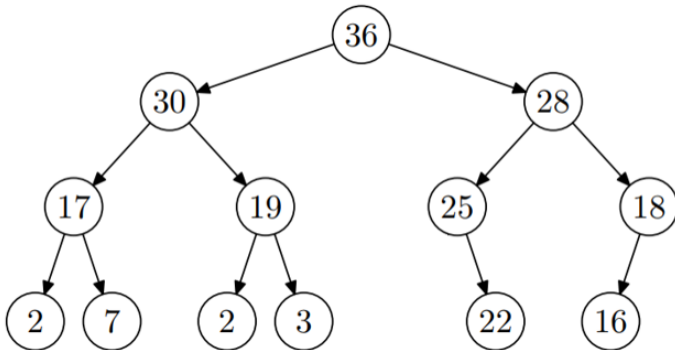
Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése

- **NEM** kupac: sérül a kupac szerkezet



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

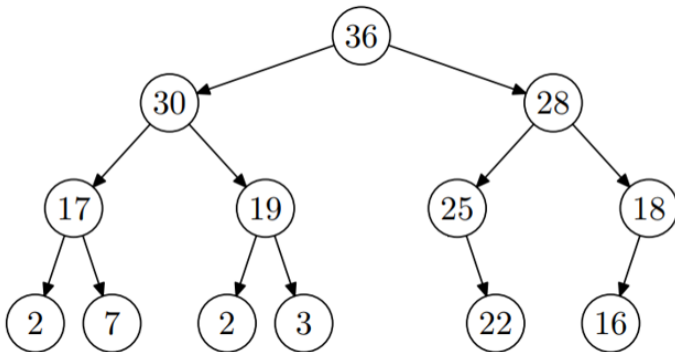
Műveletek

Alkalmazások

Elsőbbségi sor

## Törlés: legnagyobb / legkisebb elem törlése

- **NEM** kupac: sérül a kupac szerkezet
- **NEM** jó a fenti törlés "algoritmus"



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupacol:

- Legyen  $i$ . egy elem indexe
- Az  $i$ -dik elem baloldali és jobboldali részfái kupacok
- Az  $i$ . elem kisebb mint a gyerekei, vagyis sérül a kupactulajdonság

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupacol:

- Legyen  $i$ . egy elem indexe
- Az  $i$ -dik elem baloldali és jobboldali részfái kupacok
- Az  $i$ . elem kisebb mint a gyerekei, vagyis sérül a kupactulajdonság
- **Kupacol művelet:**
  - Visszaállítja a kupactulajdonságot
  - Az  $i$ -dik elemet lefele *görgeti* a kupacban

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupacol:

- Legyen  $i$ . egy elem indexe
- Az  $i$ -dik elem baloldali és jobboldali részfái kupacok
- Az  $i$ . elem kisebb mint a gyerekei, vagyis sérül a kupactulajdonság
- **Kupacol művelet:**
  - Visszaállítja a kupactulajdonságot
  - Az  $i$ -dik elemet lefele *görgeti* a kupacban

## Megjegyzés

- A kupacol műveletet felhasználhatjuk a maximum elem törlésénél is.

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupacol:

**Kupacol( $i$ )**

//  $i$  csomópont,  $A$  kupacot tároló tömb

- Ha  $A[i]$  levél  $\Rightarrow$  művelet vége

// a levél egyelemű kupac

-  $j = \text{bal\_gyerek}(i)$ ,  $k = \text{jobb\_gyerek}(i)$

-  $m = \text{maximum}(i, j, k)$

// maximum értékű elem a szülő

// és a két gyerek közül

- Ha  $m = i \Rightarrow$  művelet vége

// ha a szülő a legnagyobb, akkor teljesül

// a kupactulajdonság

-  $\text{Csere}(A[i], A[m])$

// különben felcseréljük a szülőt

// a nagyobb gyerekével

-  $\text{kupacol}(m)$

// tovább lépünk a következő szintre

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Törlés: legnagyobb / legkisebb elem törlése

- Maximum/minimum: gyökérelem
- Másoljuk az utolsó elemet a gyökér helyére
- Visszaállítjuk a kupac tulajdonságot: **kupacol(1)**:

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Törlés: legnagyobb / legkisebb elem törlése

- Maximum/minimum: gyökérelem
- Másoljuk az utolsó elemet a gyökér helyére
- Visszaállítjuk a kupac tulajdonságot: **kupacol(1)**:
  - Amennyiben a kupac tulajdonság sérül, szintenként fel kell cserélni a szülőt a nagyobb kulcsú gyerekével
  - A műveletek eredményeképpen a tulajdonságot sértő elem "lesüllyed" a helyére



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

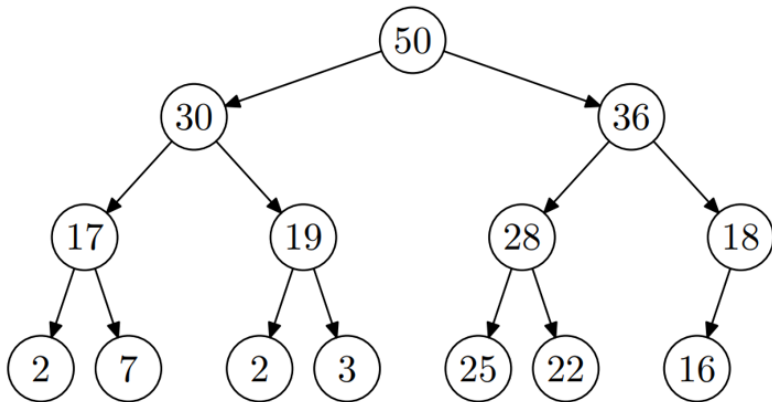
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

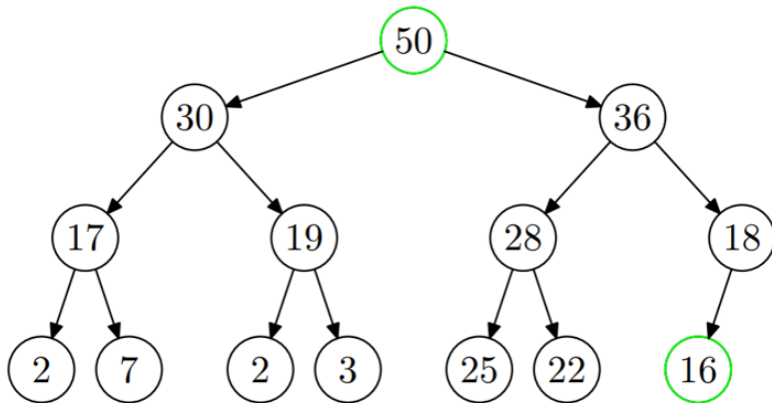
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

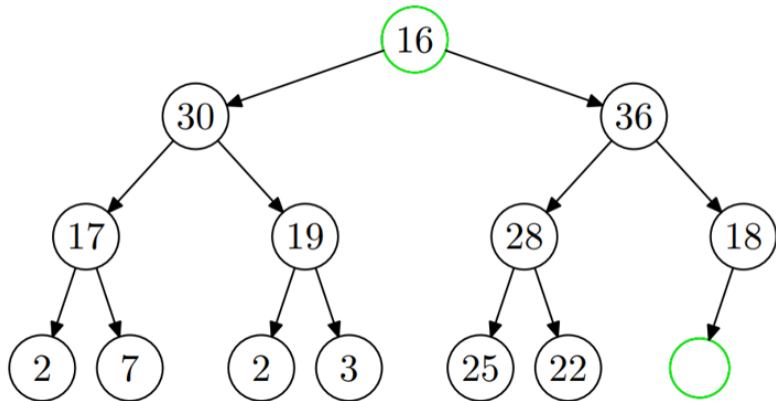
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

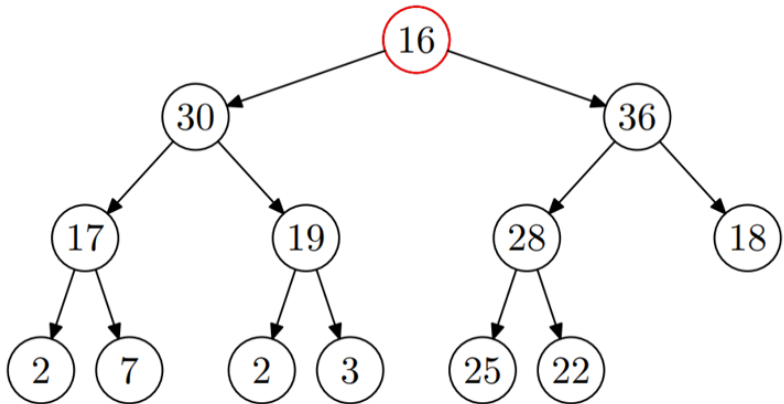
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

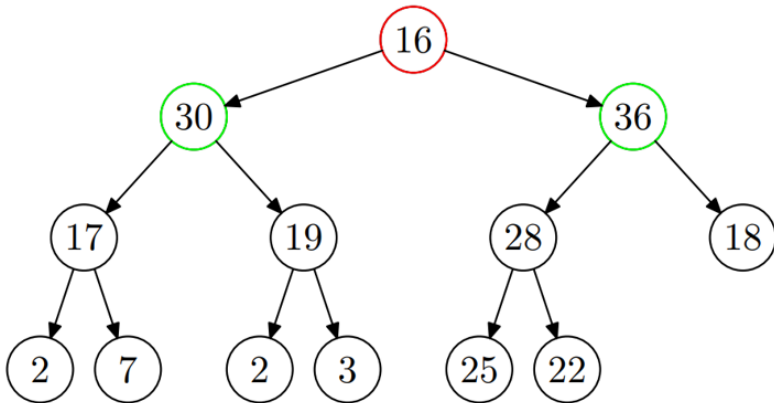
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

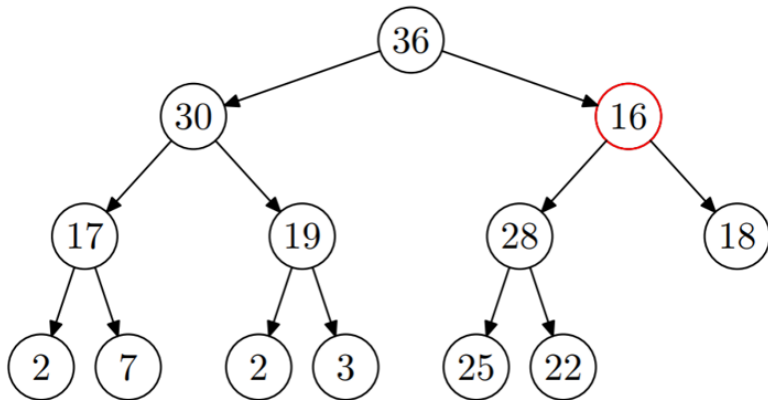
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

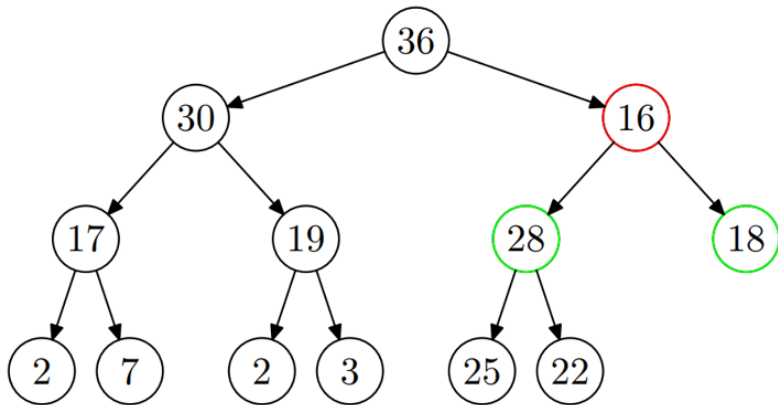
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

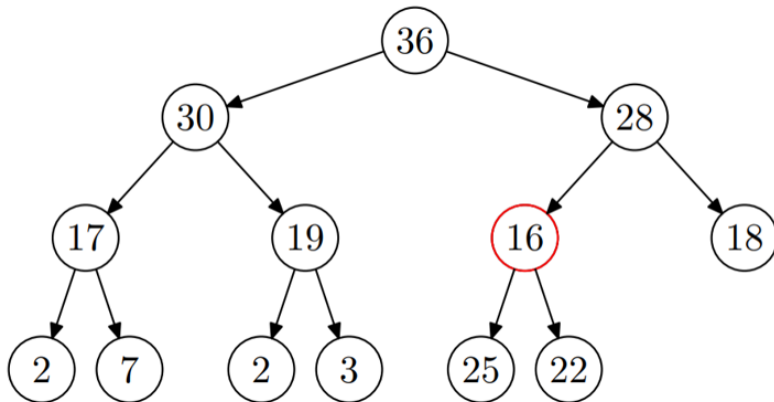
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése





# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

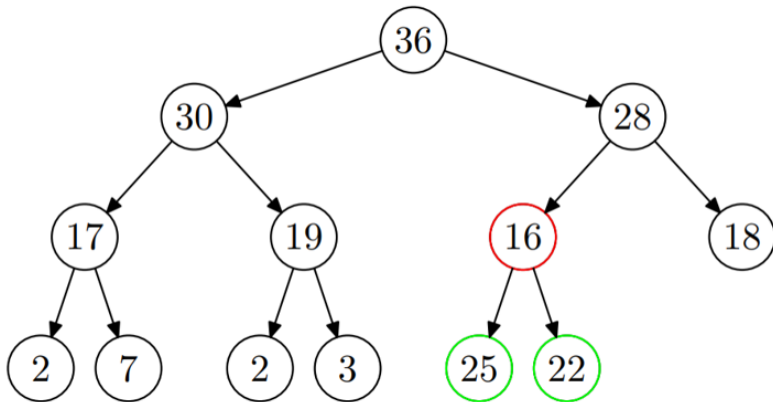
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

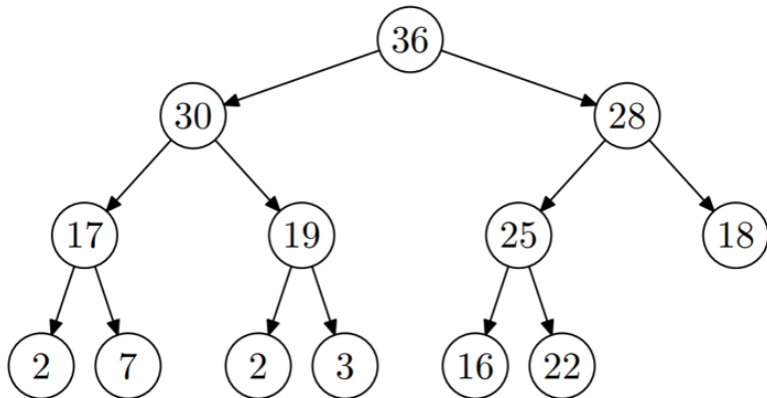
Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

**Törlés:** legnagyobb / legkisebb elem törlése



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Törlés: legnagyobb / legkisebb elem törlése

- Maximum/minimum: gyökérelem
- Másoljuk az utolsó elemet a gyökér helyére -  $O(1)$
- Visszaállítjuk a kupac tulajdonságot: **kupacol(1)**:  $O(\log n)$ 
  - Amennyiben a kupac tulajdonság sérül, szintenként fel kell cserélni a szülőt a nagyobb kulcsú gyerekével
  - A műveletek eredményeképpen a tulajdonságot sértő elem "lesüllyed" a helyére

## Futási idő: összesítve

- Futási idő:  $O(\log n)$

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupac építése 1

- A kupacba beszúrjuk valamennyi elemet.

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupac építése 1

- A kupacba beszúrjuk valamennyi elemet.

## Futási idő: összesítve

- Futási idő:  $O(n \log n)$

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

**Műveletek**

Alkalmazások

Elsőbbségi sor

## Megjegyzés

- A kupacol művelet segítségével egy tetszőleges tömb kupaccá alakítható.

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Megjegyzés

- A kupacol művelet segítségével egy tetszőleges tömb kupaccá alakítható.

## Kupac építése 2.

- Lentről felfele haladunk
  - A leveleket egyelemű kupacnak tekintünk
  - Minden köztes csomópontra meghívjuk a **kupacol** eljárást → beállítjuk a kupactulajdonságot az  $i$ . csomópont gyökerű kupacra
  - A lenről felfele sorrend biztosítja, hogy amikor egy  $i$ . csomópontra hívjuk meg a **kupacol** algoritmust, az  $i$  gyerekei már kupacot alkotnak.

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Megjegyzés

- A kupacol művelet segítségével egy tetszőleges tömb kupaccá alakítható.

## Kupac építése 2.

- Lentről felfele haladunk
  - A leveleket egyelemű kupacnak tekintünk
  - Minden köztes csomópontra meghívjuk a **kupacol** eljárást → beállítjuk a kupactulajdonságot az  $i$ . csomópont gyökerű kupacra
  - A lentről felfele sorrend biztosítja, hogy amikor egy  $i$ . csomópontra hívjuk meg a **kupacol** algoritmust, az  $i$  gyerekei már kupacot alkotnak.

## Futási idő: összesítve

- Futási idő:  $O(n)$  (bizonyításhoz ld. Cormen, Leiserson, Rivest, Stein *Új algoritmusok*)



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kupac építése 2.:

**Kupacot Épít ( $A$ ,  $n$ , kupac\_méret)**    //  $A$  kupacot tároló tömb,  
//  $n$  a tömb elemeinek a száma,  
// *kupac\_méret* a kupac végső mérete

- Kupac\_méret =  $n$     // Alapértelmezetten a kupac mérete  
// a tömb méretével egyenlő

- Minden  $i = n/2, 1$     // az első nem levél csomóponttól a gyökérig

    Kupacol ( $i$ )

- Vége(Minden)

- Vége(algoritmus)

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

**Műveletek**

Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$$n = 9$$

$$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$$

# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

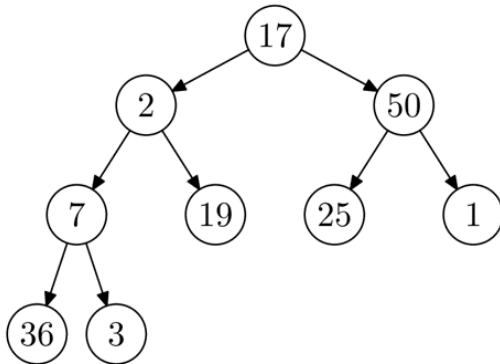
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

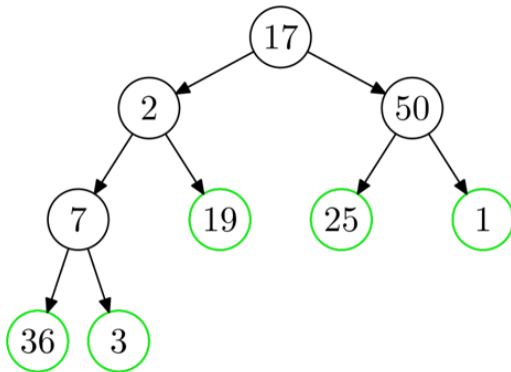
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

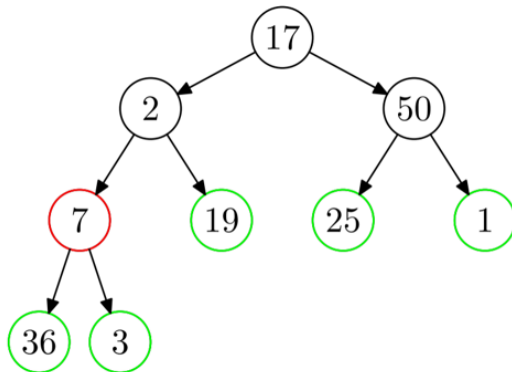
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

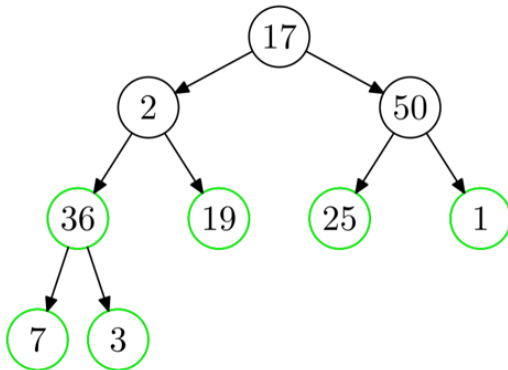
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 36, 19, 25, 1, 7, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

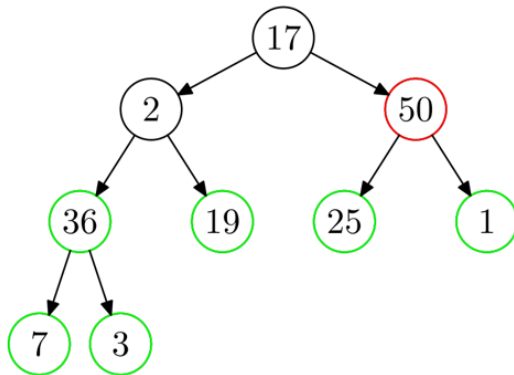
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 36, 19, 25, 1, 7, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

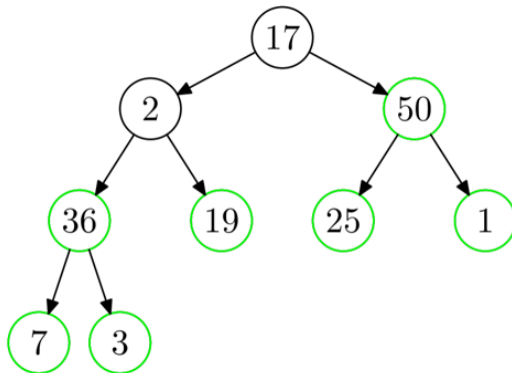
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 36, 19, 25, 1, 7, 3\}$





# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

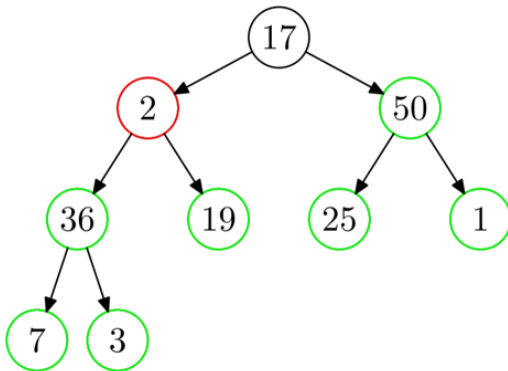
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 2, 50, 36, 19, 25, 1, 7, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

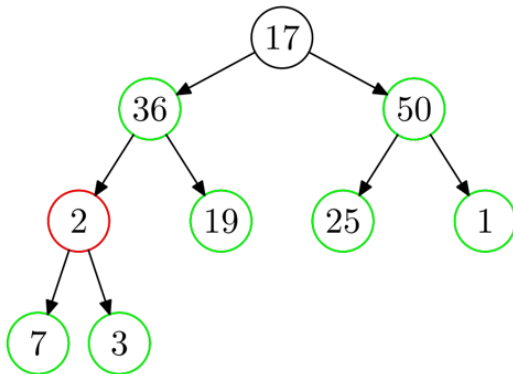
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 36, 50, 2, 19, 25, 1, 7, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

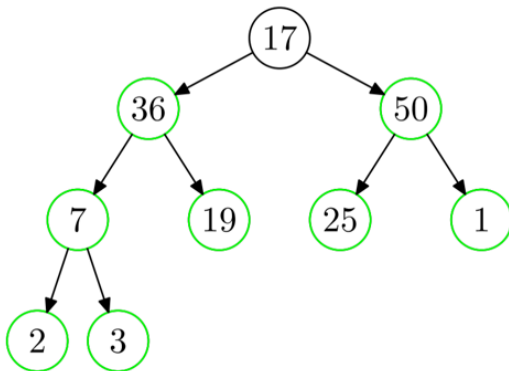
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 36, 50, 7, 19, 25, 1, 2, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

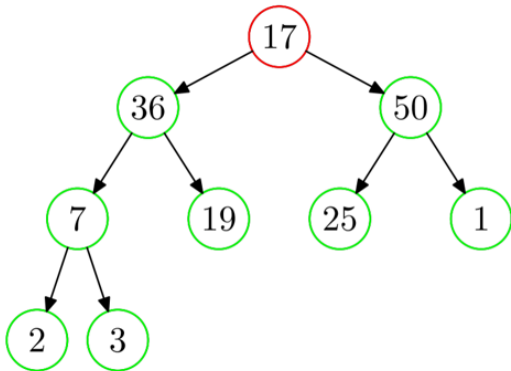
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{17, 36, 50, 7, 19, 25, 2, 7, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

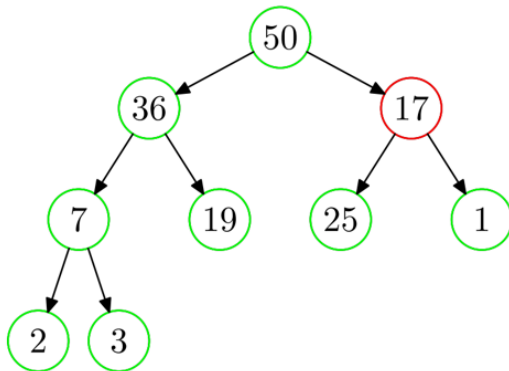
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{50, 36, 17, 7, 19, 25, 1, 2, 3\}$



# Kupac: műveletek

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

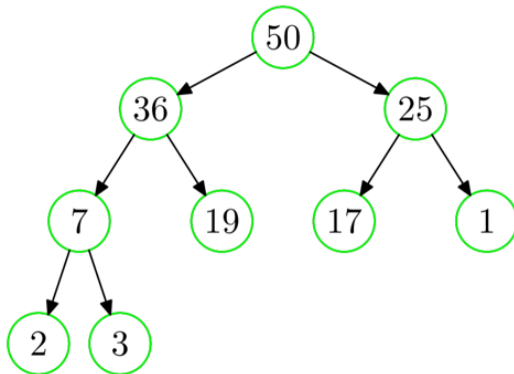
Alkalmazások

Elsőbbségi sor

## Kupac építése 2. - Példa

$n = 9$

$t = \{50, 36, 25, 7, 19, 17, 1, 2, 3\}$



## Megjegyzés

- Kupacot akkor érdemes használni, ha a feladat során gyakran van szükségünk minimum- vagy maximum keresésre.

## Példa

- Kiválasztásos rendezés:
  - Minden lépésben megkeressük a maximumot
  - **Futási idő:**  $O(n^2)$

## Megjegyzés

- Kupacot akkor érdemes használni, ha a feladat során gyakran van szükségünk minimum- vagy maximum keresésre.

## Példa

- Kiválasztásos rendezés:
  - Minden lépésben megkeressük a maximumot
  - **Futási idő:**  $O(n^2)$
- Kupacrendezés:
  - Hozzunk létre egy maximum-kupacot a bemeneti tömbből
  - Minden lépésben keressük meg és töröljük ki a maximum elemet.
  - **Futási idő:**  $n$  lépés, egy lépés  $O(\log n) \Rightarrow O(n \log n)$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Példa

- Kiválasztásos rendezés:
  - Minden lépésben megkeressük a maximumot
  - **Futási idő:**  $O(n^2)$
- Kupacrendezés:
  - Hozzunk létre egy maximum-kupacot a bemeneti tömbből
  - Minden lépésben keressük meg és töröljük ki a maximum elemet.
  - **Futási idő:**  $n$  lépés, egy lépés  $O(\log n) \Rightarrow O(n \log n)$

## Példa

- Kiválasztásos rendezés:
  - Minden lépésben megkeressük a maximumot
  - **Futási idő:**  $O(n^2)$
- Kupacrendezés:
  - Hozzunk létre egy maximum-kupacot a bemeneti tömbből
  - Minden lépésben keressük meg és töröljük ki a maximum elemet.
  - **Futási idő:**  $n$  lépés, egy lépés  $O(\log n) \Rightarrow O(n \log n)$

## Kérdés

- Lehet-e jobban/gyorsabban?

# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Példa

- Kiválasztásos rendezés:
  - Minden lépésben megkeressük a maximumot
  - **Futási idő:**  $O(n^2)$
- Kupacrendezés:
  - Hozzunk létre egy maximum-kupacot a bemeneti tömbből
  - Minden lépésben keressük meg és töröljük ki a maximum elemet.
  - **Futási idő:**  $n$  lépés, egy lépés  $O(\log n) \Rightarrow O(n \log n)$

## Kérdés

- Lehet-e jobban/gyorsabban? **NEM**

# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

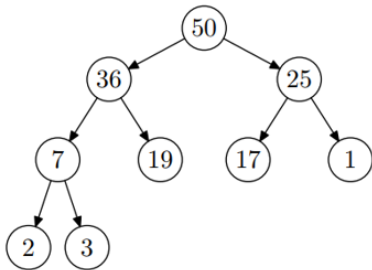
Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$

- Felépítjük a maximum-kupacot.



50	36	25	7	19	17	1	2	3
----	----	----	---	----	----	---	---	---

# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

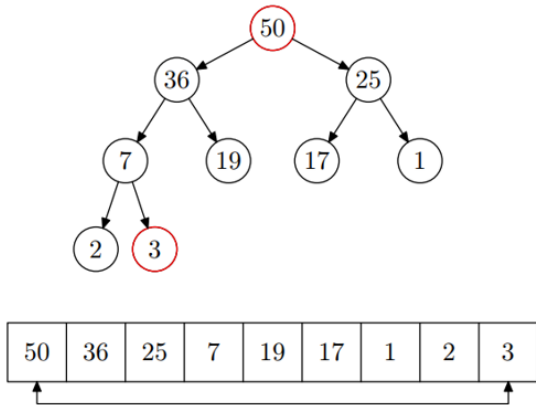
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

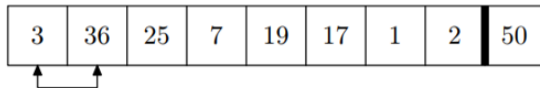
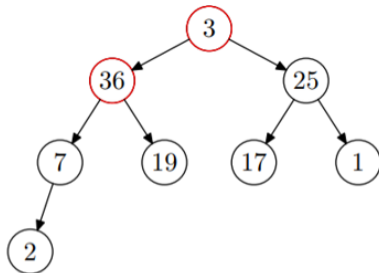
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

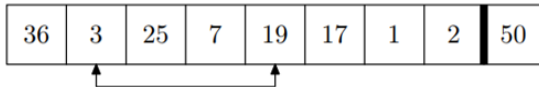
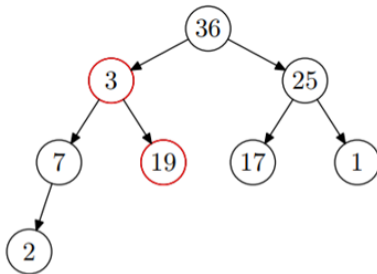
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

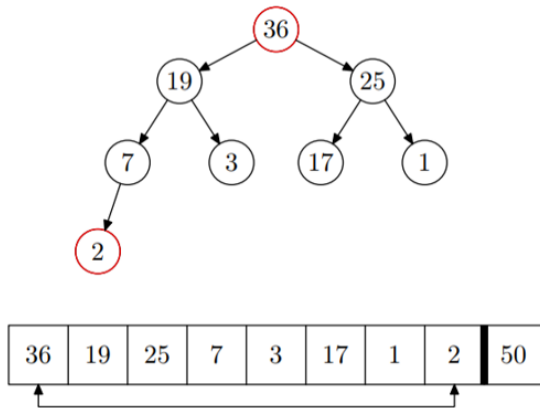
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$





# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

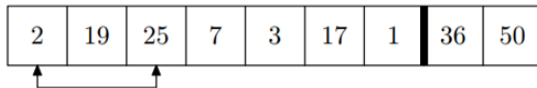
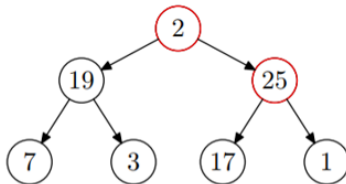
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

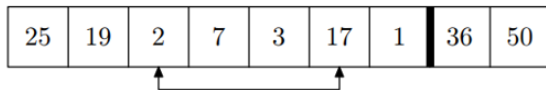
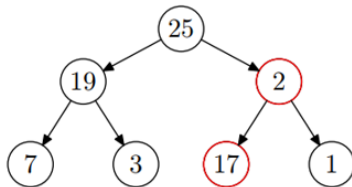
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

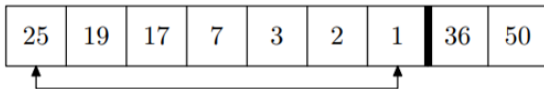
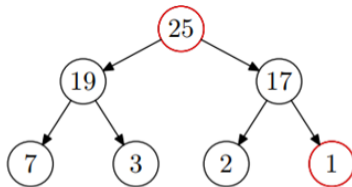
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

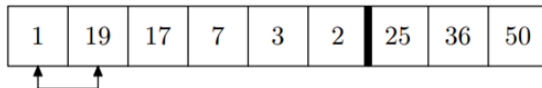
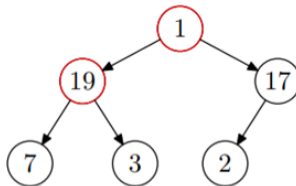
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

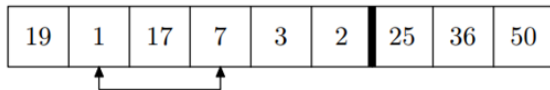
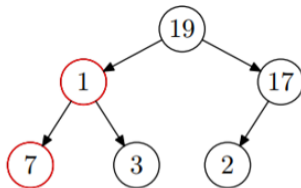
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

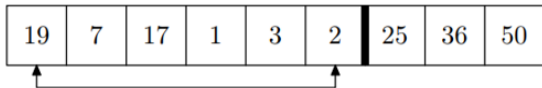
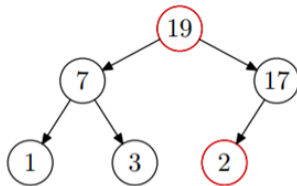
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

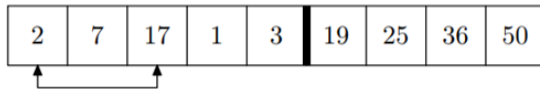
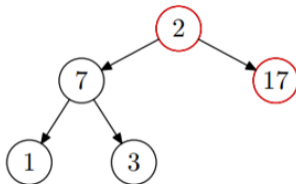
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

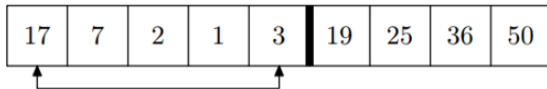
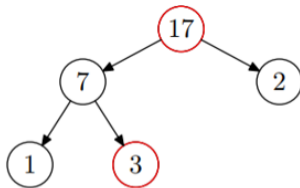
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$





# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

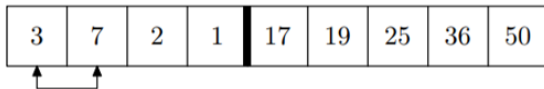
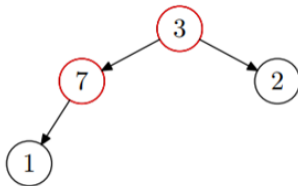
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

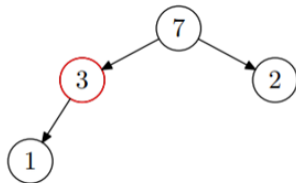
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



7	3	2	1		17	19	25	36	50
---	---	---	---	--	----	----	----	----	----

# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

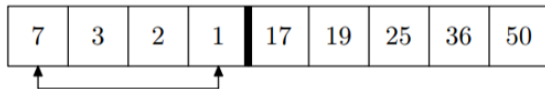
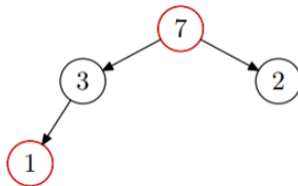
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

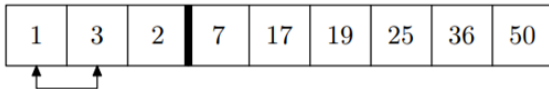
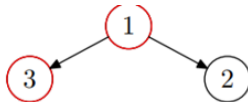
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

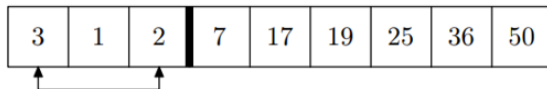
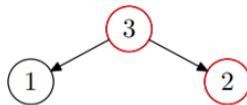
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

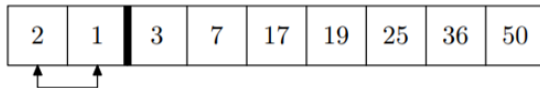
Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$



# Kupacrendezés

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Példa

$n = 9$

$t = \{17, 2, 50, 7, 19, 25, 1, 36, 3\}$

(1)

1	2	3	7	17	19	25	36	50
---	---	---	---	----	----	----	----	----

# Elsőbbségi sor AAT (*priority queue*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Feladatok ütemezése

- Az elvégzendő feladatokat és relatív prioritásukat egy elsőbbségi sorban tárolják
- Ha egy feladat elkészült a várakozó feladatok közül a *legnagyobb prioritású* feladatot választja ki az ütemező algoritmus.
- Bármely feladatnak nőhet a prioritása.



# Elsőbbségi sor AAT (*priority queue*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Feladatok ütemezése

- Az elvégzendő feladatokat és relatív prioritásukat egy elsőbbségi sorban tárolják
- Ha egy feladat elkészült a várakozó feladatok közül a *legnagyobb prioritású* feladatot választja ki az ütemező algoritmus.
- Bármely feladatnak nőhet a prioritása.

## Elsőbbségi sor

### Maximum-kupac

- **elemek:** a feladat objektumai
- **kulcs:** prioritás

# Elsőbbségi sor AAT (*priority queue*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Esemény-vezérelt szimuláció

- a prioritási sor elemei a szimulálandó események
- minden eseménynek adott a bekövetkezési időpontja
- az eseményeket a bekövetkezési idejüknek megfelelő sorrendben kell szimulálni, mivel egy esemény bekövetkezése egy másik esemény jövőbeli bekövetkezését okozhatja

# Elsőbbségi sor AAT (*priority queue*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Esemény-vezérelt szimuláció

- a prioritási sor elemei a szimulálandó események
- minden eseménynek adott a bekövetkezési időpontja
- az eseményeket a bekövetkezési idejüknek megfelelő sorrendben kell szimulálni, mivel egy esemény bekövetkezése egy másik esemény jövőbeli bekövetkezését okozhatja

## Elsőbbségi sor

### Minimum-kupac

- **elemek**: az esemény objektumai
- **kulcs**: bekövetkezési idő

# Elsőbbségi sor AAT (*priority queue*)

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Elsőbbségi sor

- Az elsőbbségi sor leghatékonyabb implementációja a minimum/maximum-kupac.

# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Medián keresése

- **Adott:** egy  $x_1, \dots, x_n$  sorozat, amelyet elemenként kapunk meg
- **Feladat:** Minden  $i$ . lépésben határozzuk meg az  $x_1, \dots, x_i$  sorozat mediánját.
- Megengedett futási idő:  $O(\log i)$ .

# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Medián keresése

- **Adott:** egy  $x_1, \dots, x_n$  sorozat, amelyet elemenként kapunk meg
- **Feladat:** Minden  $i$ . lépésben határozzuk meg az  $x_1, \dots, x_i$  sorozat mediánját.
- Megengedett futási idő:  $O(\log i)$ .

## Megoldás

- Használjunk két kupacot
  - $K_{kis}$  - maximumkupac
  - $K_{nagy}$  - minimumkupac
- Minden lépésben mindkét kupacban  $\approx i/2$  elem legyen.

# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Megoldás

- Használjunk két kupacot:
  - $K_{kis}$  - maximumkupac
  - $K_{nagy}$  - minimumkupac
- Minden lépésben mindkét kupacban  $\approx i/2$  elem legyen.
- Minden  $i$ . Lépésben:
  - Ha  $x_i \leq \max(K_{kis})$ , szúrjuk be  $x_i$ -t a  $K_{kis}$  kupacba
  - Különben (ha  $x_i \geq \min(K_{nagy})$ ) szúrjuk be  $x_i$ -t a  $K_{nagy}$  kupacba
  - Ha egyik kupacban több mint egy elemmel több van mint a másikban: tegyük át a gyökérelemet a másikba.

# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Megoldás

- Használjunk két kupacot:
  - $K_{kis}$  - maximumkupac
  - $K_{nagy}$  - minimumkupac
- Minden lépésben mindkét kupacban  $\approx i/2$  elem legyen.
- Minden  $i$ . Lépésben:
  - Ha  $x_i \leq \max(K_{kis})$ , szúrjuk be  $x_i$ -t a  $K_{kis}$  kupacba
  - Különben (ha  $x_i \geq \min(K_{nagy})$ ) szúrjuk be  $x_i$ -t a  $K_{nagy}$  kupacba
  - Ha egyik kupacban több mint egy elemmel több van mint a másikban: tegyük át a gyökérelemet a másikba.

## Eredmény

- **Medián:**  $\max(K_{kis})$  és/vagy  $\min(K_{nagy})$



# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Gráfalgoritmusok

- Dijkstra algoritmus
- Kruskal algoritmus

# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Gráfalgoritmusok

- Dijkstra algoritmus
- Kruskal algoritmus

## Egyéb

- $k$ . legkisebb elem meghatározása

# Kupac: alkalmazások

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Gráfalgoritmusok

- Dijkstra algoritmus
- Kruskal algoritmus

## Egyéb

- $k$ . legkisebb elem meghatározása

## Megjegyzés

- A kupac használható minden olyan feladatnál, ahol sorozatos minimum vagy maximum számításra van szükség.

Adatszerkezetek

Vekov Géza

Kupac

Ábrázolás

Műveletek

Alkalmazások

Elsőbbségi sor

## Kérdések

Kérdések? 😊