

ADATSZERKEZETEK ÉS ALGORITMUSOK

Keresések hatékonysága
„Hierarchikus adatszerkezetek, keresési fák”

Keresések

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

$$c_1 \log_2 n + c_2 n$$

- vagyis:

$$c_2 n$$

domináns

- Minden elem hozzáadása a rendezett tömbhöz: $\mathcal{O}(n)$

- Lehet-e a rendezett tömböt olcsóbb beszúrásokkal karbantartani?

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.