Algoritmusok és adatszerkezetek 2.

Fekete István előadása alapján
Készítette: Nagy Krisztián

2. előadás

19.1 Rendezés lineáris időben

„Edényrendezés (folytatás)"

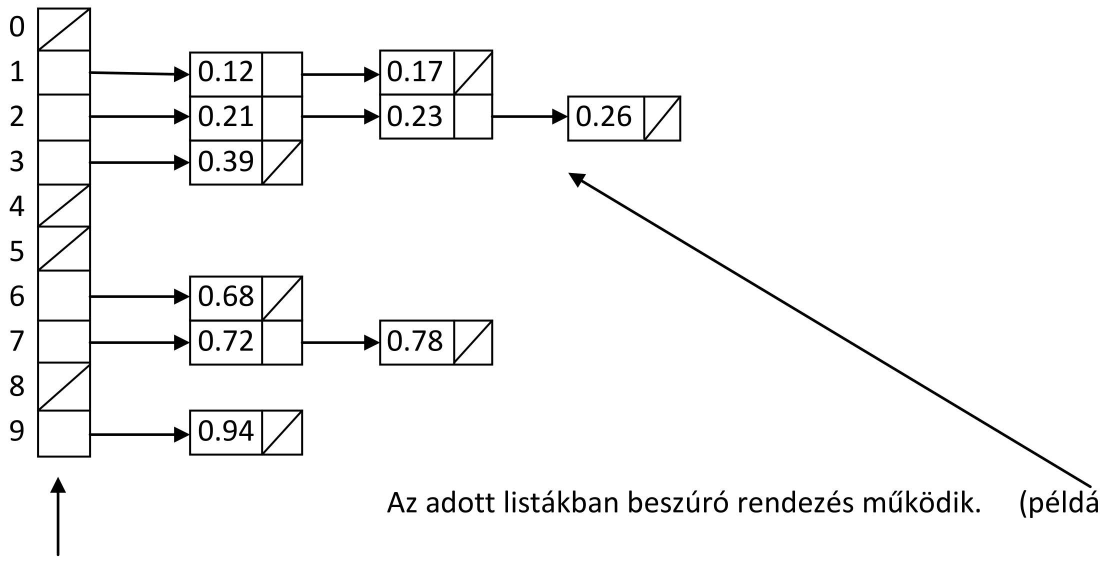
2. Edényrendezés

Input: n darab $k \in [0,1)$-beli véletlen (random) szám egyenletes eloszlásból.

Szemléltető feladat:

Legyen először az $n = 10$. Rendezd az alábbi elemeket!
$A[1..n]$: $0.78, 0.17, 0.39, 0.26, 0.72, 0.94, 0.21, 0.12, 0.23, 0.68$

T Ez a szeszélyes eloszlás alig valószínű.



Az adott listákban beszúró rendezés működik. (például itt)

Mindegyikben egy lista fejeleme foglal helyet.

Hash-függvénye: $h(r) = \lfloor 10 \cdot r \rfloor$

Elemek elhelyezése beszúrórendezéssel.

Matematikai elemzés $\Longrightarrow$ gyakorlati tapasztalat: Az egyes listák várható hossza közel 1.
Továbbá a beszúró rendezés várható összehasonlítási száma az egyes listákon közel 2.
Pontosan: $2 - \frac{1}{n}$.

Az edényrendezés három lépése:

1. $h(r)$ hash-függvény alkalmazása: $\theta(n)$
2. Edényrendezés: $\theta(n) \cdot \theta(2) = \theta(n)$
3. Output előállítása (listák összefűzése): $\theta(n)$

Tehát a teljes rendezés: $\theta(n)$
$T_{ER}(n) = \theta(n)$

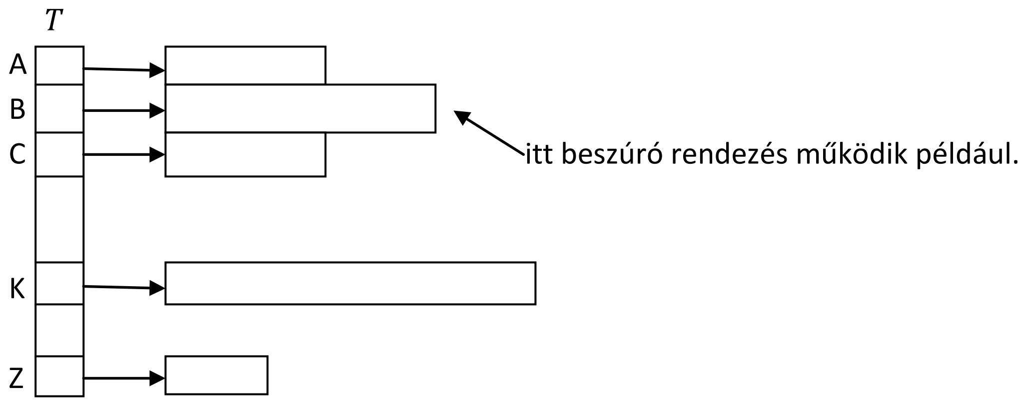
Általában tetszőleges n-re a $h(r) = \lfloor n \cdot r \rfloor$ és $T[0..n-1]$ (0-tól $n-1$-ig indexelt)
Például: n lehetne 100,1000 stb..

A fentebbi konstrukciót $(T(\quad), ábra)$ hasító táblának (hash table) nevezik.

<u>Nem jó ötlet</u> a hasító tábla m méretét n-nél kisebbre venni. Például $n = 1000$ esetén $m = 100$. Ez utóbbi konstrukció 10 hosszú listákat eredményez. Oka: A beszúró rendezés négyzetes műveletigénye nagyobb részt kap.

<u>Jó ötlet</u> viszont a másik oldalról, egy n^2 vagy $n \log n$-es rendezés oldaláról behozni a hasítás ötletét.

Például: 300 hallgatót szeretnénk névsorba rendezni.



Jobb mint, ha az egész inputot beszúró rendezéssel (vagy Quick Sort) rendeztük volna.

20. Hasítás (hash coding, hashelés)

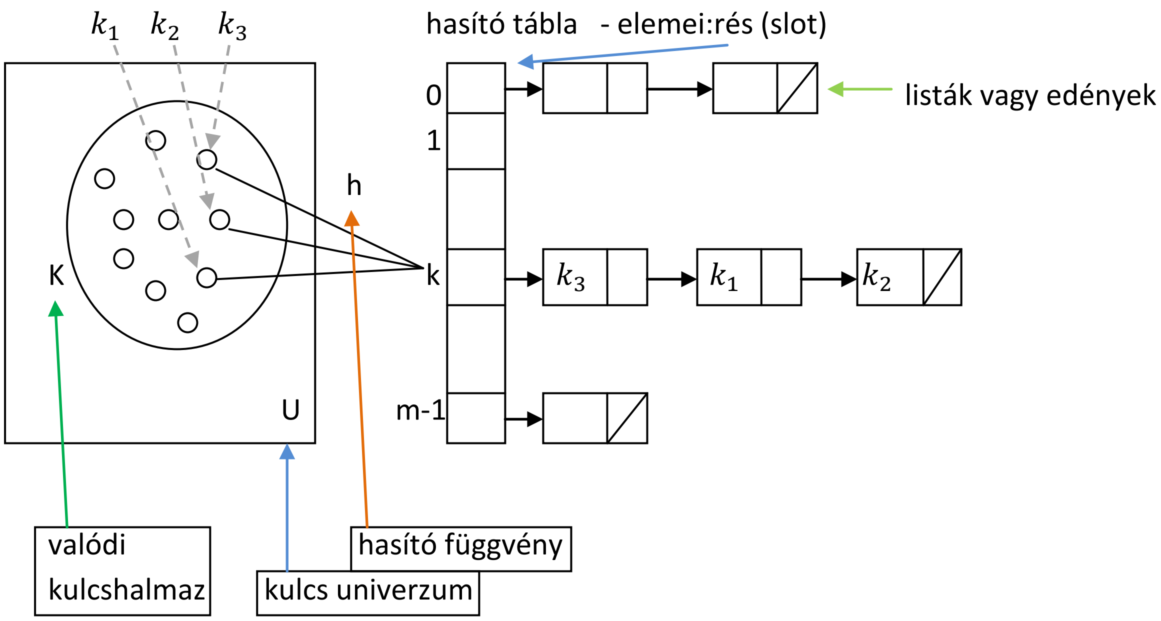
- Kulcsos rekordok hatékony elhelyezése

Eddig: Bináris keresőfa, AVL-fa, 2-3 fa (B-fa) –t ismerjük, amik kulcselhelyezésre használhatók.

Példa feladat: Adottak régi típusú személyi számok, helyezzük el őket hatékonyan!
1 890219 2806 (,de lehetett volna 2-vel kezdődő is akár ☺)

<u>Ötlet:</u> Megfelelő számú fix rekord lefoglalása.
Számoljunk: $2 \cdot 10^2 \cdot 12 \cdot 30 \cdot 10^3 \approx 74000000$ Ez oriási redundancia a 10000000 rekorddal szemben.
Megoldás: **Hasítás**

1.Hasítás láncolással



$h: U \to [0..m-1]$

$m < n < |U|, de\ lehet\ m \ll n \ll |U|\ , ahol\ n = |K|$ (kulcshalmaz elemszáma)

Kulcsütközés:
Például: $h(k_1) = h(k_2) = h(k_3)$
<u>Elvárás h-tól</u>:

- könnyen számolható legyen
- egyenletesen hasítson U-n (K-n)

Műveletek

Az egyes listák általában nem rendezettek. (Új kulcsot a lista elejére tesszük 1. rekordként)

- Beszúrás → 1. rekordként : $\theta(1)$
- Keresés → Lineáris keresés 2 $O(k - adik\ lista\ hossza) = \theta\left(\frac{n}{m}\right)$ egyenletes hasító függvény esetén
- Törlés ~ Beszúrás

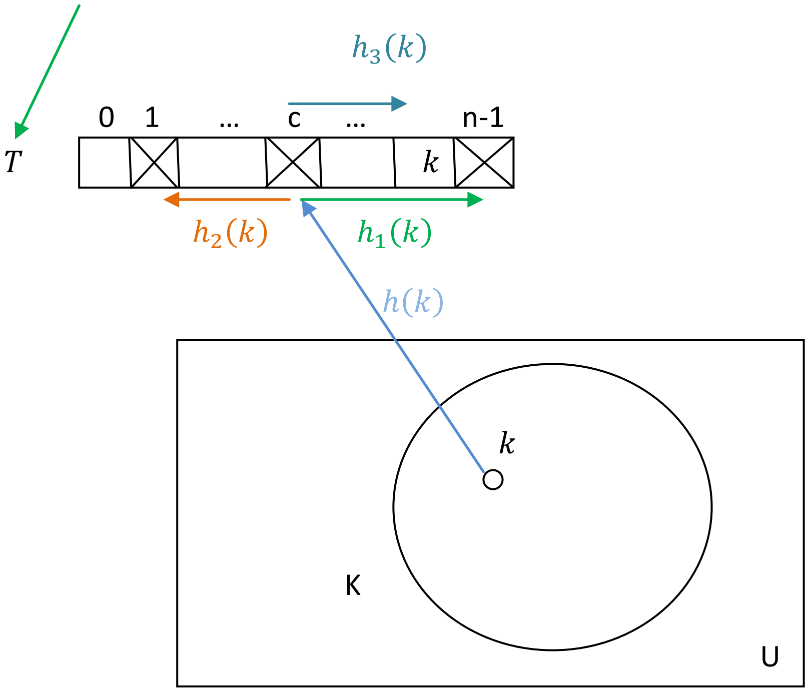
***m* megválasztása:**

$\frac{n}{m} \approx \log n \ \rightarrow m = \frac{n}{\log n}$ vagy ennél nagyobb: $m \geq \frac{n}{\log n}$. Például $n = 1000$ esetén m-et legalább 100-nak kell választani.

Így leszünk versenyképesek az AVL-fával.

2.Kulcsütközés feloldása nyitott (nyílt) címzéssel

A rekordok elhelyezésére szolgáló tömb



Az elhelyezés képlete:

$h(k) + h_i(k) \quad (mod\ m) \quad 0 \leq i \leq m - 1$
$h_0(k) \equiv 0$ (kiinduló elhelyezkedés)
Elvárás: $\{h_0(k), h_1(k), \ldots h_{m-1}(k)\} = \{0,1,2, \ldots, m - 1\}$

Mezők státusza:

a) szabad
b) foglalt
c) törölt → kereséshez elengedhetetlen

A kulcsütközés feloldásának módszerei:

1. Lineáris próbálás
2. Négyzetes próbálás
3. Kettős hasítás → nagyjából egyenletes eloszlástú

(Ezzel kapcsolatos mintafeladatokat – lásd 1. gyakorlati jegyzet)