19. Hasításos technikák (hash-elés)

Példák:

1. Ha egy telefon előfizetőket a telefonszámaikkal azonosítjuk, mint kulcsokkal, akkor egy ritkán kitöltött kulcstartományhoz jutunk. A telefonszám tehát nem alkalmas kulcsnak. Vidéki körzetben 6 jegy ⟹ ≈ 1 millió kulcsérték.
2. A személyi szám viszont nem alkalmas kulcsnak, ugyanis, ha összeszámoljuk az összes kitöltési lehetőséget, akkor ≈ 74 millió kulcsértéket kapunk.

1 790523 2806

$2*10^2*12*31*10^3$ ≈ 74 millió (Az utolsó 4 jegyből csak 3 értékes, 1. a többiből det. képződik)

A 10 millió lakosra pedig elég egy kb.: 12 millió rekordot tartalmazó file. A személyi szám tehát nem alkalmas kulcsnak.

A hash-elés jelensége:

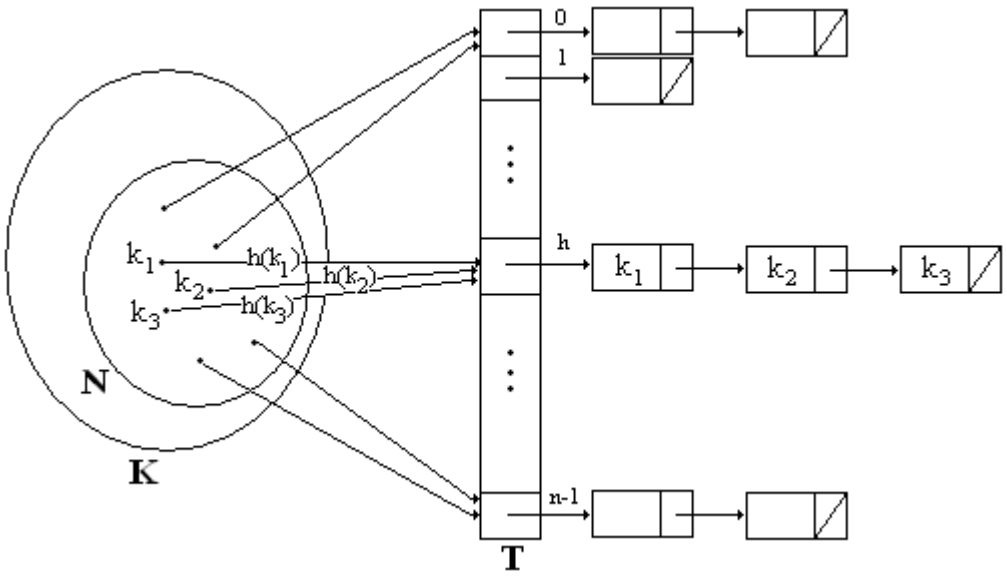
Adott a K kulcsok halmaza. A K elemivel azonosított rekordok száma azonban várhatóan jóval kisebb, mint a K számossága. Ekkor K-t egy alkalmas h függvénnyel leképezzük az ábrázolás alapját képező kisebb tartományra. Legyen ez a [0..M-1] intervallum.
A h: K→ [0..M-1] függvényt hash függvénynek nevezzük.

Mivel M< |K| , sőt általában M<<|K|, ezért h nem lehet injektív, hanem szükségképpen fellép a kulcsütközés: van olyan k ≠ k' , amelyekre h(k) = h(k').

A hash-elés ábrázolás (hasításos technikák) feladata éppen az, hogy megoldást adjanak a kulcsütközésre.

1.Hash-elés láncolással

A) Központi memóriában: a rekordok tárolása a következő ábra szerinti sémában történik.



A T hasító tábla elemei (="rései") a fejelemek, rendre azokra a listákra mutatnak, amelyek az adott résre leképezett kulcsokat tartalmazzák. Pl.: $h(k_1)=h(k_2)=h(k_3)=k$.

Az N halmazbeli kulcsok azok, amelyek a listában tárolásra kerülnek. Ha |N| = n, akkor $\alpha = \frac{n}{M}$ -et nevezzük a kitöltöttségi aránynak. Ha h egy jó hash függvény, azaz egyenletesen szórja szét a kulcsértékeket, akkor az összes lista kb. egyforma hosszú (feltéve még, hogy az input adatok kulcsai is véletlenszerűen érkeznek).
A gyakorlatban így merül fel a tárolással kapcsolatban a kérdés, hogy ha ismerjük K szerkezetét és a várható n elemszámot, akkor mekkora M-et és milyen h függvényt célszerű választani. (Ezzel később foglalkozunk).
A listákkal kapcsolatban eldöntendő kérdések:
- egy- vagy kétirányú láncolást alkalmazzunk-e?
- a listák rendezetlenek vagy rendezettek-e?
- rendezetlen esetben az elejére szúrjuk be az új elemet, vagy a végére?

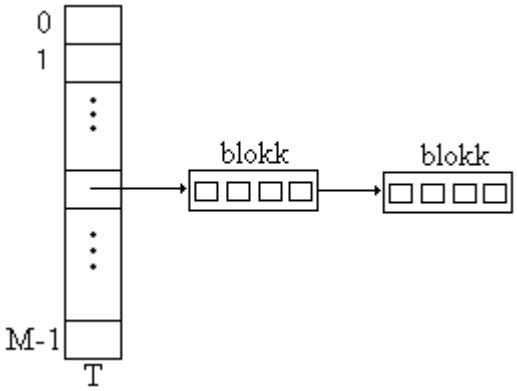
Műveletek:

Beszúrás: h(k) kiszámítása és beszúrás elsőnek a h(k)-adik listába, vagy utolsónak, ha előbb végignézzük, hogy ott van-e már az elem.

Keresés: h(k) kiszámítása és keresés a h(k)-adik listában. A sikeres keresés átlagosan l/2 összehasonlítást igényel. A sikertelen keresés átlagosan l összehasonlítást igényel.

Törlés: h(k) kiszámítása után keresés a h(k)-adik listában. Ha ott van az elem, akkor átlagosan l/2 összehasonlítást igényel.

B) Háttértárolón



A T táblázat is a háttértárolón van. A lista elemei blokkok.
1 blokk k 512 byte, kis k-ra pl. 1024 v. 2048 byte.
1 blokkban több rekord foglal helyet.

Általában a listák kevés számú blokkból állnak. Javasolt többlet ráhagyás 20-25%.
Példa:

N = 1.000.000 rekord
1 blokkban 5 rekord fér el.
Ha átlagosan 1 hosszú listákat szeretnénk, akkor M=200.000.
Legyen a rátartás miatt M=240.000.
Ekkor a keresés várhatóan 1+1 blokkelérést igényel, ha a T megfelelő részét tartalmazó blokkot közvetlenül (direkt) tudunk nyúlni.

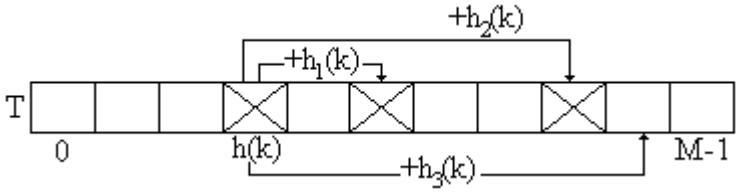
A tárolás néhány kérdése:
- lapokon (=blokkokon) belül láncoljuk-e a rekordokat?
- túlcsordulási blokkok létesítése; túlcsordulás kezelés?

Összefoglalóan:

- h, T szerepe: hatékony elérhetőség;
- a listák szerepe: kulcsütközés kezelésére.

2. Nyitott (nyílt) címzés

A belső memóriában a T[0..M-1] tömbben legfeljebb néhány ezer rekord elhelyezésére kidolgozott módszer. Természetesen legfeljebb M rekord tárolható.

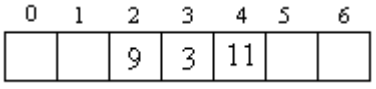


Az ábra a kulcsütközés feldolgozásának általános módszerét szemlélteti. Ha a h(k) hely foglalt, akkor szisztematikusan próbálkozunk. Az i-edik lépésben $h_i(k)$-val lépünk odább az eredeti helyről, ha kell.
A $h_i(k)$ lépésköz egyszerűbb esetben nem függ k-tól, még egyszerűbb esetben mindig 1 mezővel lépünk arrébb, mint mindjárt látjuk.
A próbálkozást a következő képlet fejezi ki:
$h(k)+ h_i(k) \pmod M \qquad (0 \le i \le M-1)$
ahol $\{ h_0(k), h_1(k), h_2(k),\ldots, h_{M-1}(k)\} = \{0,1,2,\ldots,M-1\}$
ugyanis így garantált, hogy a próbasorozat minden táblaelemhez eljut, ha szükséges.

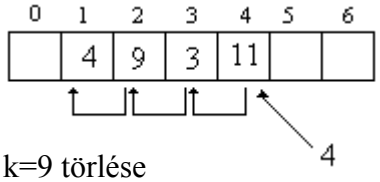
Egy mezőnek 3-féle státusza lehet:
-üres
-foglalt
-törölt (hogy a keresés folytatódhasson)

A) lineáris próbálás

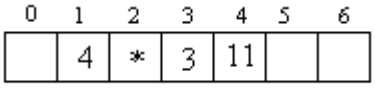
$h_i(k) = -i$
a h(k) helytől egyesével balra lépünk, amíg üres helyet nem találunk.
Pl. M = 7
h(k) = k (mod 7)
(1) k=3, 11, 9 elhelyezése:



(2) k=4 beszúrása

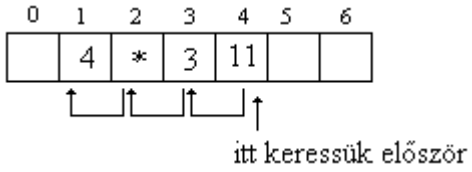


(3) k=9 törlése

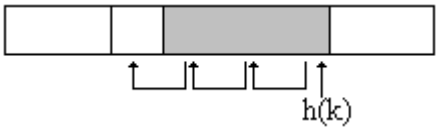


jelen esetben a * jel jelenti, hogy töröltük az elemet

(4) k=4 keresése



Hátrány: clustereződés vagy elsődleges csomósodás



B) négyzetes próbálás (álvéletlen / pszeudo véletlen)

A $h_i(k)$ most is k-tól független, de nem szabályos, hanem véletlenszerű. Egy lehetséges $h_i(k)$ az úgynevezett kvadratikus maradék próba.
Legyen M = 4k+3 alakú prímszám, pl. 19
A próbasorozat:

$$0^2, 1^2, -1^2, 2^2, -2^2, \dots, (\frac{M-1}{2})^2, -(\frac{M-1}{2})^2$$

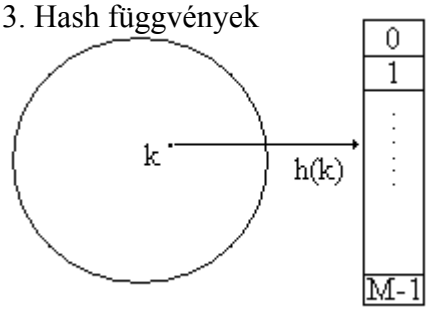
Belátható, hogy így minden értéket megkapunk [0..M-1]-ben.
Mivel $h_i(k)$ véletlenszerű, ezért nem alakul ki elsődleges csomósodás. Viszont, ha $h(k_1)=h(k_2)$, akkor k_1 és k_2 teljes próbasorozata megegyezik, ezért azonos $h(k_i)$ értékű kulcsok a közös próbasorozat mentén helyezkednek el. Ez a másodlagos csomósodás.

C) kettős hash-elés

A próbáló függvény most már függ k-tól. A h mellett egy másik h'-t is használunk.
$h(k) - i*h'(k) \pmod M$

Követelmény: h'(k) és M relatív prímek legyenek, minden k-ra.
Kettős hash-elésnél nem alakul ki 1. és 2. típusú csomósodás.
Pl. h'(k)-ra $h'(k) = k \pmod{M-1}+1$

3. Hash függvények



Milyen a jó h?
„Receptkönyv", lásd elsősorban Knuth III.
Tanács: h függjön a kulcs minden bitjétől
Elvárás: h egyenletes legyen

Osztómódszer
A k-t tekintsük egész számnak
$h(k) := k \pmod M$
M nem lehet 2 hatvány (noha egyszerűbb számítást adna)
Knuth javaslata: legyen M prím, és ne essen közel 128, 256, 512, 1024-hez.
Pl. 2000 rekord, legyen H=701, ez prím és nem esik 2 hatvány közelébe.
$h(k) = k \pmod{701}$
Tapasztalat: ez egyenletesen szór, kb. 3 elemű lánc tartozik minden kulcshoz.

Szorzómódszer:

Legyen $0 < A < 1$ rögzített

$h(k) = \lfloor M * \{kA\} \rfloor$

Lépések: k kulcs

{kA} törtrész Є [0,1)

$\lfloor M * \{kA\} \rfloor \rightarrow [0..M-1]$

M-re nem érzékeny, így legyen M alkalmas 2 hatvány.

Számolás:

Tegyük fel, hogy k elfér 32 biten

$A = \frac{\sqrt{5}-1}{2} \approx 0{,}6180$ a legjobb

(Knuth szerint)

k= 123,45

M= 10.000

h(k) = 41