

XI. Gyakorlat (2019.04.30)

Hashelés

A hashelés / hasítás célja egy olyan adattárolási eljárás megvalósítása, amelyben a keresés, beszúrás, törlés, műveletek várhatóan nagyon hatékonyak. Feltesszük, hogy a tárolandó rekordnak van egy egyedi kulcs adattagja. A kulcsok egy U halmaz elemei (U a kulcsok univerzuma), amiről ezúttal nem szükséges feltételezni, hogy rendezhető.

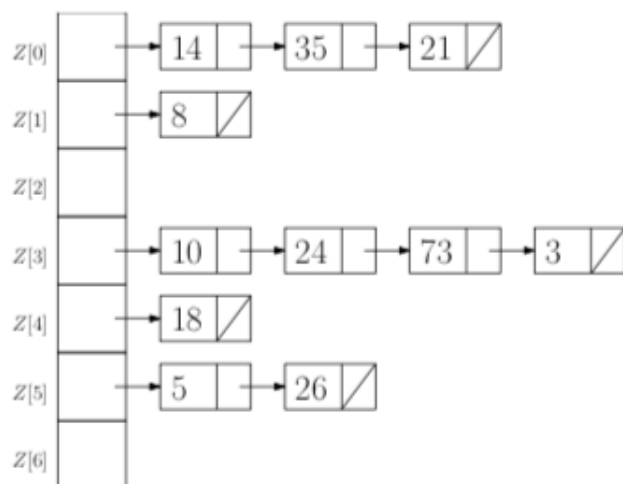
A legegyszerűbb megvalósítás az ún. direkt címzés. Ezt akkor használjuk, ha a kulcshalmaz nagyon kicsi. Ha $U = 0..m-1$, akkor egy m méretű tömbben / hasítótáblában tárolhatjuk a rekordokra mutató pointereket úgy, hogy a tömb k indexű tagja a k kulcsú rekordra mutat. Amennyiben nincs k kulcsú rekord, a megfelelő pointer értéke NULL.

Amennyiben a kulcshalmaz nagy, a direkt címzés nem alkalmazható. Ilyenkor felveszünk egy m részre osztott $Z[0..(m-1)]$ hasítótáblát, és bevezetünk egy $h:U \rightarrow 1..m-1$ hasító függvényt (tipikusan $|U| \gg m$). A k kulcsú rekordot a hasítótábla $Z[h(k)]$ részében próbáljuk meg eltárolni. Ha két adat k_1 és k_2 kulcsára $h(k_1) = h(k_2)$, akkor kulcsütközésről beszélünk. A különféle hasítási eljárások a h függvény megválasztásában és kulcsütközések kezelésében különböznek egymástól.

Láncolt / vödrös hashelés

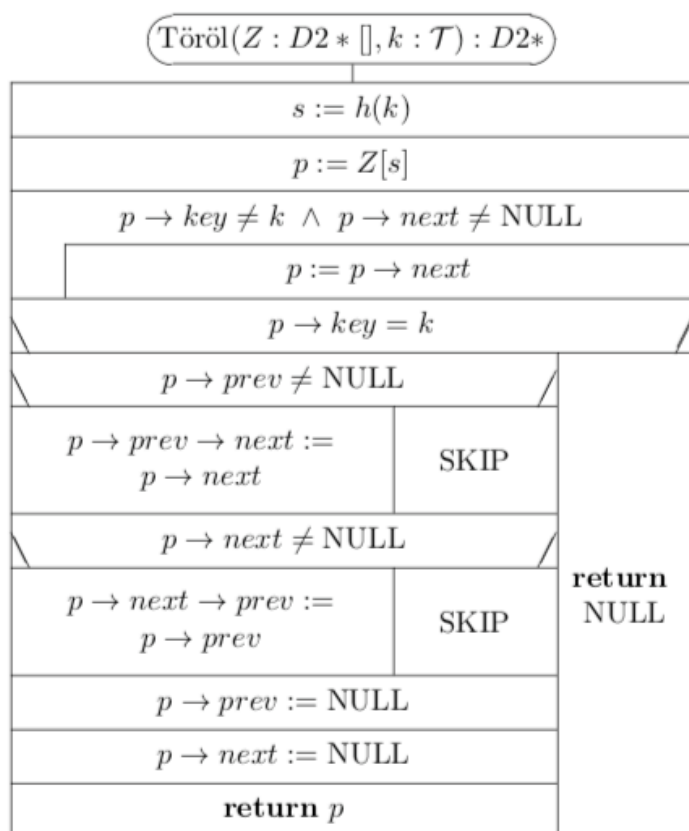
Az adatokat m darab SLL listában tároljuk. A hasítótábla $Z[i]$ eleme az i . láncolt lista első elemére mutató pointer. Ez a lista azokat a k kulcsú elemeket tartalmazza, amikre $h(k) = i$. Beszúráskor a megfelelő lista első elejére szúrjuk be az új elemet azért, hogy a beszúrás konstans műveletigényű legyen.

Példa: $m = 7$, $h : \mathbb{N} \rightarrow 0..6$, $h(k) = k \bmod 7$. A táblába beszúrt elemek sorban: 3, 21, 73, 8, 24, 35, 26, 14, 5, 10, 18.



Feladat: készítsük el a láncolt hashelés kétirányú listás változatát, és rajzoljuk fel a törlés művelet struktogramját. A kétirányú lista nem ciklikus.

A hashelés láncolt megvalósításánál a korábban tanult E1, E2 adattípusok helyett a D1 és D2 adattípusokat használjuk. Ezek azonban mindenben megegyeznek a korábbi változatukkal.



Nyílt címzéses hashelés

Ebben az esetben nem engedjük meg azt, hogy a hastábla egy eleméhez egynél több rekord is tartozzon. Ekkor a rekordokattárolhatjuk közvetlenül a hashtáblában, azonban foglalkoznunk kell a kulcsütközésekkel. Ennek érdekében a kulcsokhoz most már nem egyetlen hash értéket vezetünk be, hanem egy m -tagú sorozatot, ahol m a hashtábla mérete. Legyen $h: U \times 0..(m-1) \rightarrow 0..m-1$ egy olyan függvény, amelyre teljesül, hogy tetszőleges k kulcsra a $h(k, 0), h(k, 1) \dots h(k, m-1)$ sorozat a $0..m-1$ számok egy permutációja. h -t hasító/hash függvénynek nevezzük, az előbbi sorozatot pedig k próbasorozatának hívjuk. Általában olyan próbasorozatot választunk, amely az első eleméből önnyen rekonstruálható. Az első elemet $h_1(k)$ -val is jelölhetjük.

Beszúrás: ha a k kulcsú elemet akarjuk beszúrni a táblába, akkor először megnézzük, hogy szabad-e a $h(k, 0)$ indexű pozíció. Amennyiben igen, beszúrjuk, ha nem, akkor megnézzük szabad-e a $h(k, 1)$ pozíció. És így tovább, egészen addig, amíg nem találunk neki egy szabad helyet.

Törlés: egy elem törlése során először megkeressük az előző eljárással a helyét. Fensot, hogy a törlésnél nem változtathatjuk a megtalált pozíciót üresre, hiszen ez gondot okozhatna a későbbi kereséseknél. Épp ezért a törölt elemek helyét valójában nem szabadítjuk fel, hanem az addig ott tárolt kulcsot lecseréljük egy U halmazban szereplő D (deleted) szimbólumra.

Lineáris próba

Lineáris próba esetén a próbasorozat egy számtani sorozatot alkot.

Tehát $h(k, n) = h(k, 0) + nd \pmod m$.

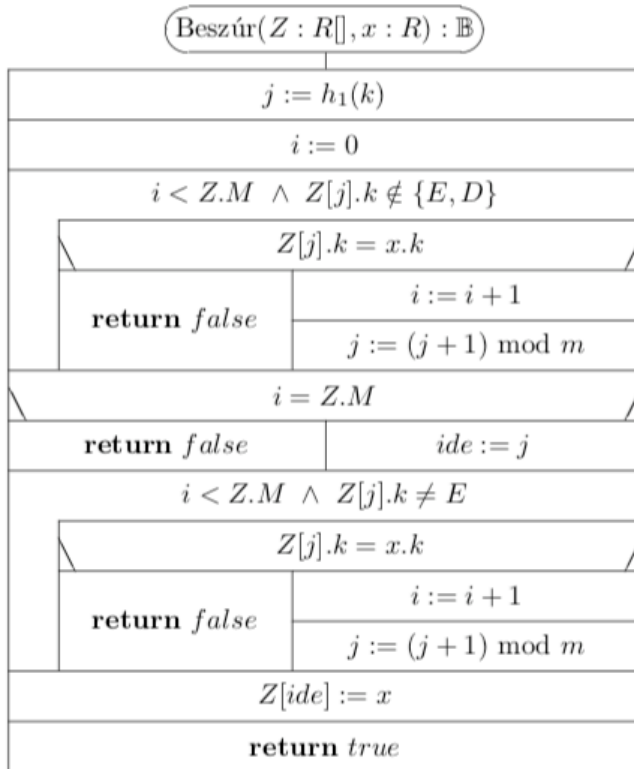
Példa: $m = 11$, $h: \mathbb{N} \rightarrow 0..10$, $h(k, n) = (k + n) \pmod{11}$.

Művelet	k	$h_1(k)$	Próbasorozat	0	1	2	3	4	5	6	7	8	9	10
Beszúr	24	2	$2_E \checkmark$			24								
Beszúr	16	5	$5_E \checkmark$			24			16					
Beszúr	57	2	$2, 3_E \checkmark$			24	57		16					
Beszúr	32	10	$10_E \checkmark$			24	57		16					32
Beszúr	15	4	$4_E \checkmark$			24	57	15	16					32
Töröl	57	2	$2, 3_{57} \checkmark$			24	D	15	16					32
Beszúr	21	10	$10, 0_E \checkmark$	21		24	D	15	16					32
Keres	2	2	$2, 3_D, 4, 5, 6_E \times$	21		24	D	15	16					32
Beszúr	2	2	$2, 3_D, 4, 5, 6_E \checkmark$	21		24	2	15	16					32
Beszúr	2	2	$2, 3_2 \times$	21		24	2	15	16					32
Keres	21	10	$10, 0_{21} \checkmark$	21		24	2	15	16					32

A lépések lejátszása után határozzuk meg a hashtábla kitöltöttségi arányszámát:

$$\alpha = \frac{6}{11}$$

Feladat: beszúrás struktogramja. A hasítótáblába illesztett adatok típusa R (rekord), ami rendelkezik egy k nevű kulcs adattaggal. Az algoritmus visszatérési értéke egy logikai érték, ami megmondja, hogy sikeres volt-e a beszúrás.



Négyzetes próba

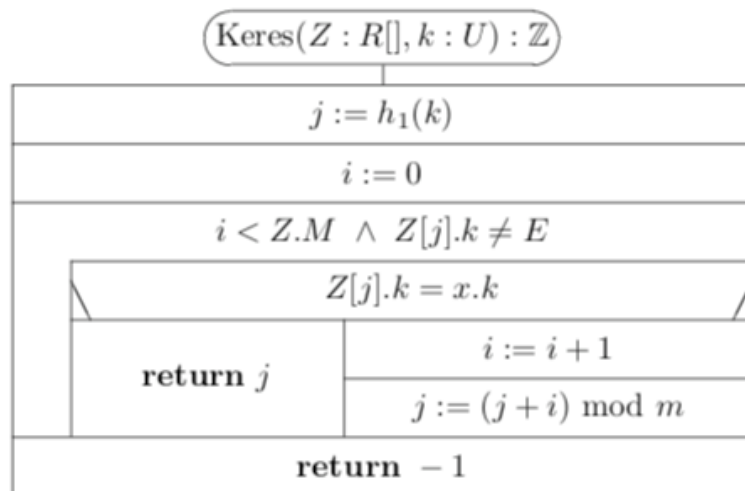
Négyzetes próba esetén a próbasorozat egy másodfokú függvény segítségével írható le, vagyis $h(k, n) = (h(k, 0) + c_1 * n + c_2 * n^2) \bmod m$ valamilyen c_1, c_2 konstansokra. (Ezeket a konstansokat úgy kell megválasztani, hogy a próbasorozat a teljes táblát kiadja, pl. kettőhatvány m esetén a $c_1 = c_2 = \frac{1}{2}$ egy jó választás, ha pedig m egy $4k + 3$ alakú prím, akkor bármilyen c_2 jó lesz $c_1 = 0$ esetén.) $c_1 = c_2 = \frac{1}{2}$ esetén $h(k, n) = h(k, n - 1) + n$.

Példa: $m = 8$, $h : \mathbb{N} \rightarrow 0..7$, $h(k, n) = (k + \frac{n+n^2}{2}) \bmod 8$.

Művelet	k	$h_1(k)$	Próbasorozat	0	1	2	3	4	5	6	7
Beszúr	13	5	$5_E \checkmark$						13		
Beszúr	20	4	$4_E \checkmark$					20	13		
Beszúr	31	7	$7_E \checkmark$					20	13		31
Beszúr	87	7	$7, 0_E \checkmark$	87				20	13		31
Beszúr	12	4	$4, 5, 7, 2_E \checkmark$	87		12		20	13		31
Töröl	31	7	$7_{31} \checkmark$	87		12		20	13		D
Keres	12	4	$4, 5, 7_D, 2_{12} \checkmark$	87		12		20	13		D
Keres	15	7	$7_D, 0, 3_E \times$	87		12		20	13		D
Beszúr	4	4	$4, 5, 7_D, 2, 6_E \checkmark$	87		12		20	13		4
Töröl	10	2	$2, 3_E \times$	87		12		20	13		4
Beszúr	35	3	$3_E \checkmark$	87		12	35	20	13		4
Beszúr	10	2	$2, 3, 5, 0, 4, 1_E \checkmark$	87	10	12	35	20	13		4

$$\alpha = \frac{7}{8}$$

Feladat: készítsük el a keresés struktogramját. Az algoritmus visszatérési értéke egy egész szám, ami megadja a paraméterül kapott kulcsú rekord indexét. Sikertelen keresés esetén a visszatérési érték -1.



Kettős hasítás

Ebben az esetben a próbasorozat egy számtani sorozat, azonban a differencia kulcsenként eltérő. Az első elemet egy $h_1 : U \rightarrow 0..m - 1$, a számtani sorozat differenciáját pedig egy $h_2 U \rightarrow 0..m - 1$ függvény adja meg. Ezekből a h hasítófüggvényt a $h(k, n) = (h_1(k) + n * h_2(k)) \bmod m$ képlet adja meg.

Példa: $m = 11$, $h_1(k) = k \bmod 11$, $h_2(k) = 1 + (k \bmod 10)$.

Művelet	k	$h_1(k)$	$h_2(k)$	Próbasorozat	0	1	2	3	4	5	6	7	8	9	10
Beszúr	23	1	4	$1_E \checkmark$		24									
Beszúr	42	9	3	$9_E \checkmark$		24								42	
Beszúr	31	9	2	$9, 0_E \checkmark$	31	24								42	
Beszúr	110	0	1	$0, 1, 2_E \checkmark$	31	24	110							42	
Beszúr	55	0	6	$0, 6_E \checkmark$	31	24	110				55			42	
Töröl	55	0	6	$0, 6_{55} \checkmark$	31	24	110				D			42	
Keres	72	6	3	$6_D, 9, 1, 4_E \times$	31	24	110				D			42	
Beszúr	14	3	5	$3_E \checkmark$	31	24	110	14			D			42	
Töröl	22	3	5	$0, 3, 6_D, 9, 1, 4_E \times$	31	24	110	14			D			42	
Töröl	110	0	1	$0, 1, 2_{110} \checkmark$	31	24	D	14			D			42	
Beszúr	13	2	4	$2_D, 6_D, 10_E \checkmark$	31	24	13	14			D			42	

Hasítófüggvény megválasztása

A $h : U \rightarrow 0..(m - 1)$ függvény egyszerű egyenletes hasítás, ha a kulcsokat a rések között egyenletesen szórja szét, azaz hozzávetőleg ugyanannyi kulcsot képez le az m rés mindegyikére. Tetszőleges hasító függvénnyel kapcsolatos elvárás, hogy egyszerű egyenletes hasítás legyen.

Osztó módszer

Ha a kulcsok egész számok, gyakran választják a $h(k) = k \bmod m$ hasító függvényt, ami gyorsan és egyszerűen számolható, és ha m olyan prím, amely nincs közel a kettő hatványához, általában egyenletesen szórja szét a kulcsokat a $0..(m - 1)$ intervallumon. Ha pl. a kulcsütközést láncolással szeretnénk feloldani, és kb. 2000 rekordot szeretnénk tárolni $\alpha \approx 3$ kitöltöttségi aránnyal, akkor a 701 jó választás: a 701 ui. olyan prímszám, ami közel esik a $2000/3$ -hoz, de a szomszédos kettőhatványoktól 512-től és az 1024-től is elég távol van.

Kulcsok a $[0,1)$ intervallumon

ha egyenletesen oszlanak el, a $h(k) = \lfloor k * m \rfloor$ függvény is kielégíti az egyszerű, egyenletes hasítás feltételét.

Szorzó módszer

Ha a kulcsok valós számok, tetszőleges $0 < A < 1$ konstanssal alkalmazható a $h(k) = \lfloor \{ k * A \} * m \rfloor$ hasító függvény. ($\{x\}$ az x törtrésze.) Nem minden lehetséges konstanssal szór egyformán jól. Knuth az $A = \frac{\sqrt{5}-1}{2} \approx 0.168$ választást javasolja,

mint ami a kulcsokat valószínűleg szépen egyenletesen fogja elosztani a rések között. Az osztó módszerrel szemben előnye, hogy érzékeny a hasító tábla méretére.