Algel IX. gyakorlat

Hash!

2012. április 2.

3. **[Vizsga: 2011. június 9.] Az alábbi hash-táblát az üresből kiindulva beszúrások sorozatával kaptuk. Határozzuk meg a beszúrások összes lehetséges sorrendjét, ha a hash-függvény a $h(x) = 3x \pmod{10}$ volt és a nyitott címzésű hash-elést lineáris próbával alkalmaztuk!**

0	1	2	3	4	5	6	7	8	9
					5	19	3	33	23

Kiszámoljuk a hash-értékeket, hogy tudjuk, ki van a helyén és ki nincs.

x	5	19	3	33	23
$h(x)$	5	7	9	9	9

Látszik, hogy 5 és 23 a helyén van, míg senki más nincs. Mivel ütközés esetén a lineáris próba miatt mindig eggyel balra lépünk, ezért 23 hamarabb jött, mint 33, az pedig hasonló okokból hamarabb, mint 3, az pedig hamarabb, mint 19 (indoklás: ha nem így lett volna, akkor az adott hely szabad lett volna, és nem a jelenleg látható pozícióban lenne valaki). Az is látszik, hogy 5 nem zavar senkit, vagyis a beszúrási sorrend: $23 \to 33 \to 3 \to 19$, és 5 pedig tetszőleges.

4. **A $T[0 : M-1]$ táblában rekordokat tárolunk nyitott címzésű hashelt szervezéssel. Az ütközések feloldására lineáris próbát alkalmazunk. Tegyük fel, hogy a tábla használata során egy hibás törlés történt: egy cellából kitöröltünk egy rekordot a törlés-bit beállítása nélkül.**

   (a) **Igaz-e, hogy a hibás törlés helye mindig megtalálható?**
   Nem, pl egy elem volt a táblában, és azt töröltük.

   (b) **Adjunk hatékony (lineáris) algoritmust a tábla megjavítására! (Módosítsuk a táblát úgy, hogy megszűnjenek a hibás törlés negatív következményei!)**
   Akkor okozhat ez problémát, ha az utolsó üres hely mondjuk i, de egy i-nél kisebb pozíciójú x elemre $h(x) \geq i$. Ha ilyen helyzet nincs, akkor jó a tábla. Arra kell gondolni, hogy a táblán végighaladva mindig az utolsó üres hely érdekel minket. Induljunk el a végéről, és minden elemre számoljuk ki $h(x)$-et, és nézzük meg, hogy a nyilvántartott utolsó üresről vagy arról túlról származik-e. Ha igen, akkor máshol nem lehetett hibás törlés, hiszen ha mégsem ott lett volna, akkor a törlés előtt is rossz lett volna a tábla. Bökjük be az utolsó üresen a törlés-bitet! A tábla jó lesz, hiszen helyes törlés után is így nézne ki a tábla. Egy végigolvasásból megvoltunk, tehát lineáris az algoritmus.

5. **A $T[0 : M-1]$ táblában $2n$ elemet helyeztünk el az első $3n$ helyen egy ismeretlen hash-függvény segítségével. A táblában minden $3i$ indexű hely üresen maradt ($0 \leq i \leq n$). Legfeljebb hány ütközés lehetett, ha az ütközések feloldására**

   (a) **lineáris próbát használtunk?**

   (b) **kvadratikus próbát használtunk?**

Mindkét esetben legfeljebb n, ugyanis 2 hosszú csomóink vannak, és minden csomóban legfeljebb 1 ütközés történhetett (mindkét esetben több ütközés esetén legalább 3 hosszú lenne a csomó). n pedig lehetett is, pl lineáris próba esetén minden $3i+2$ helyre két elemet beszúrva, kvadratikus próba esetén minden $3i+1$ helyre két elemet beszúrva.

6. **[ZH: 2005. április 8.] Egy m méretű hash-táblában már van néhány elem. Adjon $O(m)$ lépésszámú algoritmust, amely meghatározza, hogy egy újabb elem lineáris próbával történő beszúrásakor maximum hány ütközés történhet.**
Nyilván akkor történik a legtöbb ütközés, ha a (ciklikus értelemben) leghosszabb folytonosan kitöltött rész legvégére próbáljuk beszúrni az új elemet. Így elindulunk a tömb elejéről, az első üres sorozat végéig elmegyünk (a ciklikusság miatt célszerű így), ez $O(n)$, utána amíg kitöltött sorozatot találunk, egy számláló értékét növeljük. Ha üres jön, akkor a számláló értékét eltároljuk (feltéve, hogy nagyobb az eddig eltároltnál), majd nullázzuk. Ha körbeértünk, akkor vége. Ez még egy végigolvasás, tehát $O(n) + O(n) = O(n)$ lépésből megvan az egész

7. **Mi a baja a $h(k) = k^2 \pmod 7$ hash-függvénynek, ha a tábla 7 méretű?**
Nem az összes lehetséges értéket veszi fel. Ki kell számolni mind a 7 lehetőséget, és látszik.

8. **Mutassuk meg, hogy nyitott címzéses hashelés és lineáris próba esetén már két kulcshoz tartozó hash-függvényérték megegyezése is okozhat tetszőlegesen nagy méretű csomósodást!**
Konstruktívan: legyen $h(x_1) = h(x_2)$, ezután $h(x_i) = h(x_{i-1}) - 1$. Így csak x_1-hez és x_2-höz tartozó érték egyezik meg, mégis látható, hogy tetszőleges nagy n-re n hosszú csomónk lesz (és minden beszúrás ütközéses volt).

9. **[ZH: 2010. április 19.] Egy M méretű hash-táblába $n < M$ elemet raktunk be nyitott címzéssel, kvadratikus próbával, a $h(x)$ hash-függvényt használva. Ennek során t_1 ütközés történt (ennyiszer kellett tovább próbálkoznunk, egy elem beszúrása során több ütközés is lehetett). Ugyanezt az n elemet ugyanabban a sorrendben beszúrtuk egy M^2 méretű hash-táblába is, de most lineáris próbával, $M \cdot h(x) + 1$ hash-függvénnyel, ekkor t_2 ütközés történt. Igazolja, hogy $t_2 \leq t_1$.**
Vegyük észre, hogy mivel az eredeti táblában a méretkorlát miatt legfeljebb M elemet tároltunk, valamint az új hash-függvény minden eredeti hash-értéket M távolságra helyez egymástól, így az új táblában a különböző hash-értékű elemek nem ütközhetnek lineáris próba esetén (ehhez legalább $M + 1$ eddigi ütközés kellett volna). Így ütközés csak azért adódhat a második esetben, mert az ütköző elemek hash-értéke megegyezik. Ezekben az esetekben viszont ugyanezek az ütközések megtörténtek az eredeti táblában is (a próbasorozatok értékfüggetlensége miatt). Tehát legalább annyi ütközés volt az első, mint a második esetben.

10. **[Vizsga: 2008. június 3.] Az 1 és 91 közötti összes 3-mal osztható egész számot valamilyen sorrendben egy M méretű hash-táblába raktuk a $h(x) = x \pmod M$ hash-függvény segítségével, lineáris próbával. Ennek során hány ütközés fordulhatott elő, ha $M = 35$, illetve ha $M = 36$?**
$M = 35$ esetén a $3 \ldots 33$ számok a saját helyükre kerülnek, $0 \pmod 3$ helyekre, $36 \ldots 69$ számok $1 \pmod 3$ helyekre, $72 \ldots 90$ számok pedig $2 \pmod 3$ helyekre, és mindegyik elfér, tehát nincs ütközés. $M = 36$-nál csak $3 \ldots 33$ fér be ütközés nélkül, $36 \ldots 69$ pont ugyanezekre a helyekre menne, $72 \ldots 90$ szintén. Innen már ki lehet számolni.

11. **[Vizsga: 2005. május 26.] A kezdetben üres M méretű hash-táblába sorban beraktuk a $k_1, k_2, \ldots, k_n$ kulcsokat a $h(x) \equiv x \pmod M$ hash-függvénnyel, lineáris próbával. Jelölje t_1 a keletkezett táblában az egymás melletti foglalt mezők maximális számát. Amikor ugyanezt a $k_1, k_2, \ldots, k_n$ sorozatot ugyanabban a sorrendben egy üres $2M$ méretű táblába rakjuk be a $h(x) \equiv x \pmod{2M}$ hash-függvénnyel, lineáris próbával, akkor a kapott táblálban legyen t_2 az egymás melletti foglalt mezők maximális száma.**

    (a) **Igazolja, hogy $t_2 \leq t_1$**
    Tfh $t_2 > t_1$. Ekkor vegyünk egy t_2 hosszú sorozatot a $2M$ méretű táblában, és nézzük meg, hogy ezek hogyan helyezkednének el az eredeti táblában (a beszúrások sorrendje megegyezik!). Vegyük észre, hogy az eredetibe beszúráskor, ha egyéb ütközés nem volt,

ugyanezt a sorozatot kapjuk (ciklikus értelemben, modulo aritmetika miatt), vagy esetleg hosszabbat is (lineáris próba miatt). Ez ellentmond a feltételnek, tehát $t_2 \leq t_1$.

(b) **Igaz-e, hogy $t_1 \leq 2t_2$?**
Ellenpélda: $|0|4|2| \pmod 3$, de $|0|\ |2|\ |4|\ | \pmod 6$, $t_1 = 3$, $t_2 = 1$, és $3 \not\leq 2 \cdot 1$.