Algel IX. gyakorlat

Hash!

2012. április 2.

1. Nyitott címzéssel hash-eltünk egy 11 elemű táblába a $h(k) = k \pmod{11}$ hash-függvény segítségével. A következő kulcsok érkeztek (a megadott sorrendben): $10, 22, 31, 4, 15, 28, 17, 88, 59$. Adjuk meg a tábla végső állapotát a következő két próbamódszerre:

   (a) lineáris próba;

   (b) kvadratikus maradék próba!

2. **[Vizsga: 2010. január 21.]** Kettős hashelést használva szúrja be egy kezdetben üres, $M = 11$ méretű táblába a következő kulcsokat (ebben a sorrendben): 26, 3, 48, 14, 15, 7. A használt hash függvény legyen
$$h(k) = (k \pmod{M}),$$
a próbasorozat hash függvénye pedig
$$h'(k) = 1 + (k \pmod{M-5}).$$
Minden beszúrás után rajzolja le a tábla pillanatnyi állapotát!

---

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

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?

   (b) Adjunk hatékony (lineáris) algoritmust a tábla megjavítására! (Módosítsuk a táblát úgy, hogy megszűnjenek a hibás törlés negatív következményei!)

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 ($0 \le i \le n$). Legfeljebb hány ütközés lehetett, ha az ütközések feloldására

   (a) lineáris próbát használtunk?

   (b) kvadratikus próbát használtunk?

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.

7. Mi a baja a $h(k) = k^2 \pmod{7}$ hash-függvénynek, ha a tábla 7 méretű?

8. Mutassuk meg, hogy nyitott címzéses hashelés és lineáris próba esetén már két kulcshoz tartozó hash-függvényérték megegyezése is okozhat tetszőlegesen nagy méretű csomósodást!

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 $t_2 \leq t_1$.

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 $M = 36$?

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 $t_2 \leq t_1$

    (b) Igaz-e, hogy $t_1 \leq 2t_2$?

12. Nyitott címzéssel, lineáris próbálással akarjuk a $K_1 < K_2 < \cdots < K_n$ kulcsú elemeket egy tömbbe hash-elni a beszúrási algoritmus következő módosításával: ha egy K kulcsú elem beszúrásának megkísérlésekor a K-nál nagyob K' kulcsú elem foglalja el a cellát, akkor a K kulcsú elemünket behelyezzük ebbe a cellába, és a beillesztést K helyett a K' kulcsú elemmel folytatjuk a következő cellánál (és ha ott a K'-nél nagyobb K'' kulcsú elemet találjuk, akkor az előbbihez hasonlóan járunk el). Bizonyítsuk be, hogy az n elem beillesztése után kapott tömb független az elemek beszúrási sorrendjétől!