Schlotter Ildi
ildi@cs.bme.hu

8. gyakorlat
Hashelés

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

2. A $b_0 ... b_n$ alakú $n+1$ hosszú bitsorozatokat akarjuk tárolni. Tudjuk, hogy a b_0 paritásbit, ami a sorozatban az egyesek számát párosra egészíti ki. Ha nyitott címzésű hash-elést használunk $h(x) \equiv x \pmod{M}$ hash-függvénnyel és lineáris próbával, akkor $M = 2^n$ vagy $M = 2^n + 1$ méretű hash-tábla esetén lesz várhatóan kevesebb ütközés?

3. 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. (Ciklikusan értve, azaz t_1 a következő beszúráskori leghosszabb próbasorozat hossza.) 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 táblában legyen t_2 az egymás melletti foglalt mezők maximális száma.
a) Igazolja, hogy $t_2 \leq t_1$.
b) Igaz-e, hogy $t_1 \leq 2t_2$?

4. A $T[0:M-1]$ táblában rekordokat tárolunk nyitott címzésű hashelt szervezéssel. Az ütközések feloldására lineáris próbálást alkalmazunk. Tegyük fel, hogy a tábla használata során egy hibás törlés történt: egy cellából kitöröltünk egy rekordot a törlés-bit beállítása nélkül. (Vagyis a cellán nem látszik, hogy töröltünk belőle.)
a) Igaz-e, hogy a hibás törlés helye mindig megtalálható?
b) Adjunk lineáris időigényű algoritmust a tábla „megjavítására”. (Módosítsuk úgy a táblát, hogy megszűnjenek a hibás törlés negatív következményei.)

5. A kakukk-hash ötlete a következő. Adott egy M méretű tábla és két hash függvény (h_1 és h_2), melyek értékkészlete $\{0, 1, \ldots, M-1\}$. A cél, hogy bármely x elem vagy a $h_1(x)$ vagy a $h_2(x)$ sorszámú cellába kerüljön. Ehhez x beszúrása során a következő módon járunk el: ha a $h_1(x)$ vagy a $h_2(x)$ sorszámú cella üres, akkor elvégezzük a beszúrást (valamelyik cellába), különben x „kilöki” ezen két cella közül az egyikből az ott lévő elemet (a két cella közül véletlenszerűen választva). Ha egy y elemet kilöknek az egyik hash függvény szerinti helyéről, akkor a másik hash-függvény által meghatározott helyére kerül, kilökve ezzel esetleg egy újabb elemet. Az így kilökött elem is hasonlóan jár el, stb.

   Adjunk algoritmust, amely detektálja, hogy a tábla egy adott kitöltési helyzeténél van-e olyan elem, melynek beszúrásakor a fenti algoritmus sohasem áll le!

Gyakorlás:

1. Nyitott címzéssel hasheltünk egy 11 elemű táblába a $h(k) = k \pmod{11}$ hash-függvény és kvadratikus maradék próba segítségével. A következő kulcsok érkeztek (a megadott sorrendben): $6, 5, 7, 17, 16, 3, 2, 14$. Add meg a tábla végső állapotát! Mit kaptunk volna, ha lineáris próbát használtunk volna?

2. A hash-függvény legyen $f(K) = K \pmod{M}$, a táblaméret $M = 7$, és $1 \leq K \leq 20$. Helyezzük el a táblában a 3, 4, 7, 11, 14, 17, 20 kulcsokat ebben a sorrendben
a) lineáris
b) kvadratikus maradék próbálást használva az ütközések feloldására.

3. Nyitott címzéssel hashelünk egy kezdetben üres $M = 11$ méretű táblába, $h(K) = K \pmod{M}$ hash-függvénnyel. Az ütközések feloldására kettős hashelést használunk, ahol a második hash-függvény $h'(K) = (7K \pmod{(M-1)})$. Mi lesz a tábla állapota, ha a 4, 5, 14, 7, 8, 26, 18 kulcsokat szúrjuk be ebben a sorrendben?

4. Előfordulhat-e nyitott címzéses hash-elés esetén, hogy az $n > 3$ méretű táblában csak 3 elem van, de a keresés lépésszáma n?

5. 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.