2-3 fák, B-fák, hashelés

Algoritmuselmélet
8. gyakorlat

2013. április 12.

1. Illesszük be az alábbi 6 kulcsot egy kezdetben üres $(2,3)$-fába a megadott sorrendben: D, B, E, A, C, F. Rajzoljuk le az eredményül kapott fát!

2. Az $[1,178]$ intervallum összes egészei egy $(2,3)$-fában helyezkednek el. Tudjuk, hogy a gyökérben két kulcs van, és az első kulcs a 17. Mi lehet a második? Miért?

3. Egy $(2,3)$-fába egymás után 1000 új elemet illesztettünk be. Mutassa meg, hogy ha ennek során egyszer sem kellett csúcsot szétvágni, akkor a beillesztések sorozata előtt már legalább 2000 elemet tároltunk a fában.

4. Egy kezdetben üres $(2,3)$-fába az $1, 2, \ldots, n$ számokat szúrtuk be ebben a sorrendben. Bizonyítsa be, hogy a keletkezett fában a harmadfokú csúcsok száma $O(\log n)$.

5. Egy B_{20}-fának (huszadrendű B-fának) 10^9 levele van. Mekkora a fa szintjeinek minimális, illetve maximális száma?

6. Adottak a sík egész koordinátájú $P_1 = (x_1, y_1), P_2 = (x_2, y_2), \ldots, P_n = (x_n, y_n)$ pontjai. Javasoljunk $O(n \log n)$ költségű módszert annak eldöntésére, hogy vannak-e olyan P_i, P_j pontok $(i \neq j)$, melyek távolsága nem több mint 2.

7. Az MSc-re jelentkezőknek a felvételit alkotó 3 témakör mindegyikéből lesz egy írásbeli pontszámuk (P_1, P_2, P_3), és keletkezik egy felvélteli pontszámuk is (FP). Tegyük fel, hogy a P_i-k 1 és 30 közötti egészek, míg az FP tetszőleges pozitív egész szám lehet. Adjon meg egy olyan adatszerkezetet, amivel a következő műveletek az adott időben végrehajthatóak (n a jelentkezők számát jelöli)!
BESZÚR(P_1, P_2, P_3, FP): az adott pontszámok beillesztése – $O(\log n)$
KERES(p): a pontosan p felvételi ponttal $(FP = p)$ rendelkező jelentkezők számát határozza meg – $O(\log n)$
KORLÁT(i, q): az írásbelin az i-edik témakörből legalább q pontot elért jelentkezők számát határozza meg – $O(1)$

8. Mi a baja az $f(K) = K^2 \pmod 7$ hash-függvénynek, ahol 7 a táblaméret?

9. A hash-függvény legyen $f(K) = K$, 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.

10. 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.

11. 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 kevesebb ütközés?

12. A kezdetben üres M méretű hash-táblába sorban beraktuk a $k_1, k_2, \ldots, k_n$ kulcsokat a $h(x) \equiv x (\bmod M)$ 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 $h(x) \equiv x (\bmod 2M)$ 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 $t_1 \leq 2t_2$?