

Keresőfák II.

- Illesszük be egy piros-fekete fába sorban a 8, 2, 4, 7, 5, 3, 1, 6, 9 elemeket. Ezután töröljük az 5, majd a 7 elemet.
 - Milyen költséggel tudunk bináris keresőfával illetve piros-fekete fával rendezni egy n elemű listát?
 - Egy sportklub teniszezői kialakítottak egy erősort. Ezt a következő szabályok szerint tartják karban: (1) új játékos mindig a sorrend végére kerül, (2) a rangsor szerinti i -edik játékos kihívhatja az $i + 1$ -ediket, és ha legyőzi, helyet cserélnek. Tervezzük olyan hatékony adatszerkezetet, mely lehetővé teszi a rangsor számítógépes kezelését! A szükséges funkciók a következők:
 BESZÚR(név): az új játékost "név" azonosítóval a rangsor végére teszi
 KIHÍV(név): az $i + 1$. helyen álló játékos nevét adja, ha a "név" azonosítója játékos az i . helyen áll és $i \geq 1$.
 KICSERÉL(i): kicseréli a rangsor i . és $i + 1$. helyén lévő játékosokat, ha $i \geq 1$.
 - Szeretnénk on-line, egyszerre maximum $15n$ darab p hosszú csomagot (bitsorozatot) puffereálni úgy, hogy közben a tárolt csomagokon bizonyos műveleteket végrehajthassunk. A csomagok rendelkeznek egy 1 és k közötti címzett azonosítóval ($k \ll n$) és egy 1 és $30kn$ közötti üzenet azonosítóval. Lehetővé szeretnénk tenni, hogy üzeneteket az egyes címzettek lekérdezhessenek vagy törölhessenek a tárból üzenetazonosító alapján $O(\log n)$ költséggel. Emellett meghatározhatjuk a számukra legrégebben küldött üzenet azonosítóját, majd lekérdezzük azt összesen $O(\log \max k, n)$ lépésben. Javasoljunk megfelelő adatstruktúrát a feladatra!
 - Mi a hasonlóság és a különbség a piros-fekete és a (2,3)-fák között?
 - Illesszük be egy B_4 fába sorban a 8, 2, 4, 7, 5, 3, 1, 6, 9 elemeket. Ezután töröljük az 5, majd a 7 elemet. Mi lett volna, ha az eredeti fából a 4 elemet szerettük volna törölni?
 - Egy kezdetben üres (2,3)-fába az $1, 2, \dots, n$ számokat szúrtuk be ebben a sorrendben (növekvő). Bizonyítsuk be, hogy a keletkezett fában a harmadfokú csúcsok száma $O(\log n)$.
 - Egy B_{20} -fának (huszadrendű B-fának) 10^9 levele van. Mekkora a fa szintjeinek minimális, illetve maximális száma?
-
- Mi a baja az $f(K) = K^2 \pmod{7}$ hash-függvénynek, ahol 7 a táblaméret?
 - 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
 - lineáris
 - kvadratikus maradék
 próbálást használva az ütközések feloldására.
 - Egy M méretű hash-táblába $n < M$ elemet raktunk be nyitott címzéssel, kvadratikus próbával. Ennek során t_1 ütközés történt, azaz ennyiszor 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. Igazoljuk, hogy $t_2 \leq t_1$! (ZH 2010. tavasz)
 - 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
 - lineáris próbálást
 - kvadratikus maradék próbálást használtunk?
 - 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. Adjuk meg a tábla végső állapotát!

14. A $b_0 \dots b_n$ 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?
 15. A kezdetben üres M méretű hash-táblába sorban beraktuk a $k_1, k_2, \dots, k_n$ 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 beszúrás kori leghosszabb próbасorozat hossza.) Amikor ugyanezt a $k_1, k_2, \dots, k_n$ 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 $t_1 \leq 2t_2$?
 16. Egy 10 méretű hash-táblába kettős hash-elést használva helyezzük el a következő elemeket: 12, 5, 20, 11, 25, 2. A használt hash-függvény legyen $h(x) = 3x \pmod{10}$, a második hash-függvény pedig $h'(x) = (x \bmod 9) + 1$. (PPZH 2010. tavasz) Mi a baj a $h'(x)$ értékeivel?
 17. A $T[0 : M - 1]$ táblában rekordokat tárolunk nyitott címzésű hashelt szervezéssel. Az ütközések feloldására lineáris próbálást alkalmazunk. Tehát ha a $h(K)$ sorszámú cella foglalt, akkor a K kulcsú rekordot a $h(K) - 1, h(K) - 2, \dots$ sorszámú cellák közül az első üresbe tesszük. 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 időigényű) algoritmust a tábla megjavítására. (Módosítsuk úgy a táblát, hogy megszűnjenek a hibás törlés negatív következményei.)
-
18. Hogyan állapíthatnánk meg egy térképet reprezentáló gráf éllistája (utak és hosszai) alapján $O(n^3)$ lépésben, hogy mi a legrövidebb, $3k$ élből álló út a gráf bármely pontjából, bármely másíkba?
 19. Hogyan tudnánk az előző feladat lépésszámán javítani, ha nem a legrövidebb utat keressük, hanem szeretnénk minél kisebb eséllyel eltévedni, azaz a legkevesebb várost/csomópontot szeretnénk érinteni. Tudunk-e jobb lépésszámkorlátot biztosítani?
 20. Adott az $a_1, a_2, \dots, a_n$ rendezetlen, valós számokból álló lista. Adjunk algoritmust, amely $O(n \log_9 n^2)$ lépésben meghatározza az összes olyan $i - j$ párt, amelyre $a_i^2 + a_j^2 = c$. (PZH 2010.05.20)
 21. A valós számokból álló $a_1, \dots, a_n$ sorozat olyan, hogy az $a_1^{2^n}, a_2^{2^n}, \dots, a_n^{2^n}$ sorozat egy darabig nő, utána csökken. Adjunk $O(n)$ összehasonlítást használó algoritmust, ami rendezi az $a_1, \dots, a_n$ sorozatot. (ZH 2007. ápr. 27.)
 22. Adott k darab n hosszú (rendezetlen) sorozat, amelyek közül bármelynek bármely részsorozatáról tudjuk, hogy az első és az utolsó elem különbségének abszolútértéke páros vagy prím szám, emellett pedig 5000-nél nem nagyobb. Készítsünk a sorozatokból egyetlen rendezett listát $O(kn)$ lépésben.
 23. Legyen $f_1(n) = n^{3 \log n}$ és $f_2(n) = 2010 \cdot 4^{\log^2 n}$. Igaz-e, hogy $f_1 = O(f_2)$, illetve, hogy $f_2 = O(f_1)$?