

- Egy 2–3 fa kezdetben csak a 6, 8, 13 elemeket tárolja. Rajzoljuk le ezt a fát, majd szűrjük be a 2, 5, 1 elemeket, végül töröljük a 8, 2 elemeket.
- A $h(x) = x \pmod{7}$ hash-függvény segítségével hajtsuk végre az alábbi hash-táblán rendre a TÖRÖL(13), KERES(7) és BESZŪR(7) műveleteket úgy, hogy az ütközések feloldására használjunk (a) lineáris próbát; (b) kvadratikus maradék próbát; (c) kettős hash-elést a $h'(x) = 1 + (x \pmod{6})$ másodlagos hash-függvénnyel.

0	1	2	3	4	5	6
0	8				12	13

- Egy különböző egész számokat tároló 2–3 fa gyökerében két útjelző van, a 101 és a 117. Legfeljebb hány elemet tárolhat a fa? (vizsga3, 2016.)
- Egy kezdetben üres, 11-méretű hash-táblába nyílt címezéssel, a $h(x) = x \pmod{11}$ hash-függvényt és lineáris próbát használva szűrünk be néhány egész számot. Két elem kitörölése után az alábbi állapotot kaptuk. Határozzuk meg az összes olyan 30-nál kisebb pozitív egész számot, amit (a) a 7-es indexű, illetve (b) a 3-as indexű cellából törölhettünk.

0	1	2	3	4	5	6	7	8	9	10
11	1	26	*	15		6	*			10

- Vázzuk a 2–3 fának (és műveleteinek) egy olyan módosítását, amiben továbbra is van KERES, BESZŪR, TÖRÖL, MIN, MAX művelet, és ezeken kívül van még RANG és K-ADIK művelet is, ahol RANG(x) azt adja vissza, hogy a tárolt elemek között az x a rendezés szerint hányadik elem, a K-ADIK(i) pedig, hogy a rendezés szerint a tárolt elemek közül melyik az i -edik. A módosítás során a felsorolt szokásos műveletek lépésszámának nagyságrendje ne változzon, és mindkét új művelet lépésszáma legyen $O(\log_2 n)$, ahol n a tárolt elemek száma. (pzh, 2008.)
 - 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?
-
- Egy 2–3 fában az 1, 5, 7, 8, 12, 13, 20, 21 elemeket tároljuk, és a levelek feletti szinten a csúcsoknak (balról jobbra haladva) 3, 3, 2 levelük van. (vizsga2, 2012. alapján)
 - Rajzoljuk fel a 2–3 fát.
 - Szűrjük be a fába a 6-ot.
 - Töröljük a fából a 8-at.
 - Egy 7-méretű hash-táblába a $h(x) = x \pmod{7}$ hash-függvénnyel szűrünk be elemeket. Az ütközéseket kettős hasheléssel oldjuk fel a $h'(x) = 5 - (x \pmod{5})$ másodlagos hash-függvény segítségével. Szűrjük be a kezdetben üres táblába a 19, 26, 38, 33, 31 elemeket ebben a sorrendben. (vizsga3, 2018.)
 - Vödörös hash-eléssel M vödörben tárolunk n elemet. Határozzuk meg a keresés lépésszámát (a legrosszabb esetben), ha a vödrök tartalmát (a) láncolt listákban, (b) 2–3 fákban tároljuk. (vizsga4, 2018. alapján)
 - Diákok pontszámait szeretnénk nyilvántartani egy többfordulós verseny során, ahol a diákok kódokkal vannak azonosítva; minden kód egy tetszőleges egész szám. Tervezzük adatszerkezetet, amiben a következő műveleteket kell tudni elvégezni.
 - BESZŪR_DIÁK(x): beilleszt az adatszerkezetbe egy új, x kódú diákot 0 ponttal;
 - PONT_NÖVELÉS(x, y): az x kódú diák pontjait y -nal megnöveli;
 - HÁNY_PONT(x): megadja, hogy az x kódú diáknak hány pontja van;
 - KI_A_LEGJOBB: visszaadja az összes olyan diák kódját, akinek a pontszáma a legmagasabb.
 A KI_A_LEGJOBB művelet lépésszáma $O(k)$ legyen, ahol k a legtöbb ponttal rendelkező diákok számát jelöli, a többi művelet lépésszáma pedig $O(\log_2 n)$ legyen, ahol n a résztvevő diákok száma. (zh2, 2023.)
 - Adottak az a_1, a_2, \dots, a_n egész számok, amikről tudjuk, hogy az a_1 a „középső”, azaz a rendezés szerinti $\lfloor n/2 \rfloor$ -edik elem, egyébként azonban a számok rendezetlenek. Ezekből építsünk fel egy adatszerkezetet, amiben az alábbi két művelet van: a BESZŪR egy új elemet illeszt az adatszerkezetbe, a KÖZÉPTÖR pedig az aktuális középső elemet törli. Az adatszerkezet kezdeti felépítése legyen $O(n)$ összehasonlítás, és ha éppen k elemet tárolunk, akkor mindkét művelet lépésszáma legyen $O(\log_2 k)$. (zh, 2007. alapján)
 - Adott egy M -méretű T tömb, aminek néhány cellájában egy-egy egész számot tárolunk, a többi cella pedig üres. Adjunk $O(M^2)$ lépésszámú algoritmust annak eldöntésére, hogy keletkezhetett-e ez a tömb úgy, hogy az elemeket a $h(x) = x \pmod{M}$ hash-függvény segítségével és lineáris próbát használva szűrjük be valamilyen sorrendben, és ha igen, akkor az algoritmus határozzon is meg egy ilyen sorrendet.