

1. Egy piros-fekete fában a 2010, 42, 100π , 1848, 3 elemeket tároljuk úgy, hogy a gyökérben levő elem a 42. Adjuk meg az összes ilyen piros-fekete fát. (zh, 2011.)
2. Egy 2–3 fa kezdetben csak a 6, 8, 13 elemeket tárolja. Rajzoljuk le ezt a fát, majd szúrjuk be a 2, 5, 1 elemeket, végül töröljük a 8, 2 elemeket.

3. Mennyi a tárolható elemek számának minimuma, illetve maximuma egy olyan piros-fekete fában, aminek a fekete-magassága 3?
4. 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.)
5. Olyan bináris fát használó adatszerkezetet szeretnénk, ahol minden csúcsban nemcsak egy elemet, hanem még egy olyan számlálót is nyilvántartunk, ami azt mondja meg, hogy az adott csúcs részfájában összesen hány darab elemet tárolunk (beleértve magát a csúcst is). Vázzuk
 - (a) a bináris keresőfának,
 - (b) a piros-fekete fának,
 illetve a műveleteiknek egy olyan módosítását, amitől a KERES, BESZÚR, TÖRÖL műveletek lépésszáma nem változik meg.
6. Vázzuk a 2–3 fának (és műveleteinek) egy olyan módosítását, amiben továbbra is van KERES, BESZÚR, 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.)

7. Egy 2–3 fában az 1, 5, 7, 8, 12, 13, 20, 21 kulcsokat tároljuk, és a levelek feletti szinten a csúcsoknak (balról jobbra haladva) 3, 3, 2 levelük van. (vizsga2, 2012. alapján)
 - (a) Rajzoljuk fel a 2–3 fát.
 - (b) Szúrjuk be a fába a 6-ot.
 - (c) Töröljük a fából a 8-at.
8. Mekkora lehet egy olyan piros-fekete fa magassága, amiben 7 elemet tárolunk? (pzh2, 2024.)
9. Adott két piros-fekete fa; mindkettőben n darab különböző elemet tárolunk. Adjunk $O(\log_2 n)$ lépésszámú eljárást, ami eldönti, hogy az első fa összes eleme nagyobb-e a második fa összes eleménél.
10. Tervezzünk adatstruktúrát a következő feltételekkel. Természetes számokat kell tárolni, egy szám többször is szerepelhet, és a szükséges műveletek a következők.
 - BESZÚR(i): az i szám egy újabb példányát tároljuk;
 - TÖRÖL(i): az i szám egy példányát töröljük;
 - MINDTÖRÖL(i): az i szám összes példányát töröljük;
 - DARAB(i): visszaadja, hogy hány példány van az i számból.
 Az adatstruktúra legyen olyan, hogy ha n -féle elemet tárolunk, akkor mindegyik művelet lépésszáma $O(\log_2 n)$. (zh, 2003.)
11. Diákok pontszámait szeretnénk nyilvántartani egy többfordulós verseny során, ahol a diákok kódokkal vannak azonosítva; minden kód egy tetszőleges egész szám. Tervezzünk adatszerkezetet, amiben a következő műveleteket kell tudni elvégezni.
 - BESZÚR_DIÁK(x): beilleszt az adatszerkezetbe egy új, x kódú diákot 0 ponttal;
 - PONT_NÖVELES(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.)
12. Egy n -csúcsú bináris fában, melyben nem tárolunk elemeket, a csúcsok mindegyike pirosra vagy feketére van színezve. Adjunk $O(n)$ lépésszámú eljárást annak eldöntésére, hogy be lehet-e írni a fa nemlevél csúcsaiba különböző valós számokat úgy, hogy ezzel egy piros-fekete fát kapjunk.