

1. (a) Építsünk beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból: 4, 3, 7, 5, 1, 8, 2, 6.
(b) Hajtsuk végre rendre a TÖRÖL(8), TÖRÖL(7) és TÖRÖL(4) műveleteket.
(c) Járjuk be pre-, in- és posztorder bejárással a kapott 5-csúcsú bináris keresőfát.
2. 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.)

3. Egy bináris keresőfában csupa különböző egész számot tárolunk. Lehetséges-e, hogy egy KERES(x) hívás során a keresési út mentén a 20, 18, 3, 15, 5, 8, 9 kulcsokat látjuk ebben a sorrendben? (zh, 2004.)
4. Lehetséges-e, hogy egy piros-fekete fából a preorder bejárás 6, 1, 5, 3, 2, 4 sorrendben olvassa ki az elemeket? (pzh, 2008.)
5. Mennyi a tárolható elemek számának minimuma, illetve maximuma egy olyan piros-fekete fában, aminek a fekete-magassága 3?
6. 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 csúcsot is). Vázoljuk
(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.
7. Egy bináris keresőfában n darab különböző egész számot, egy másikban pedig m darab különböző egész számot tárolunk. Rendezzük az $n + m$ elemet $O(n + m)$ lépésben. (vizsga2, 2007. alapján)

8. Egy bináris fa inorder bejárása $j, b, k, g, i, a, c, d, f, e, h$, preorder bejárása pedig $a, b, j, g, k, i, d, c, e, f, h$. Rekonstruáljuk a fát. (vizsga2, 2002.)
9. Mekkora lehet egy olyan piros-fekete fa magassága, amiben 7 elemet tárolunk? (pzh2, 2024.)
10. 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.
11. Adott két bináris keresőfa, melyekben összesen $2n$ darab különböző elemet tárolunk. Tudjuk, hogy az első fa magassága $\lceil 2024 \log_2 n \rceil$, a másodiké pedig $\lceil n/2024 \rceil$, továbbá az első bináris keresőfa minden eleme kisebb a második fa minden eleménél. Adjunk olyan $O(\log_2 n)$ lépésszámú algoritmust, ami előállít egy olyan bináris keresőfát, ami mind a $2n$ elemet tartalmazza. (zh2, 2024. alapján)
12. 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 lépésgénye $O(\log_2 n)$. (zh, 2003.)
13. 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.
14. Egy gyökeres színtezett fán A és B a következő játékot játssza: felváltva mozgatnak egy bábut, ami kezdetben az első szinten, a gyökérben van. Minden lépésben a soron következő játékos az aktuális v csúcsból v valamelyik fiába mozgatja a bábut. A játéknak akkor van vége, ha a bábu a fa egyik levelébe kerül. A levelek egy része zöldre van festve. A kezdő A játékos akkor nyer, ha a játék egy zöld levélben ér véget. Adott a fa éllistája, amiből minden levélről kiderül, hogy az zöld-e. Mutassunk egy $O(n)$ lépésszámú algoritmust, amely meghatározza, hogy az A játékos hogyan játsszon, hogy biztosan nyerjen (feltéve, hogy van ilyen nyerő stratégiája). (zh, 2005.)