

7. gyakorlat Piros-fekete fák, 2-3-fák, B-fák

1. Mutassuk meg, hogy $O(n)$ forgatással bármely két n csúcsú bináris fa átalakítható azonos alakúra (a forgatásokat csak az egyik fán alkalmazva)!
2. Tegyük fel, hogy T_1 és T_2 két piros-fekete fa, valamint x egy olyan szám, amely T_1 minden kulcsánál nagyobb, de T_2 minden kulcsánál kisebb. Adjunk algoritmust egy olyan piros-fekete fa létrehozására $O(\log n)$ időben, amely T_1 és T_2 elemeit, valamint x -et tartalmazza (ahol n a nagyobbik fa elemeinek száma).
3. 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?
4. Az S_1 és S_2 kulcshalmazokat kiegészített 2-3-fákban tároljuk. Ezek az eredeti 2-3-fától abban különböznek csak, hogy minden csúcsban fel van jegyezve az onnan induló részfa magassága. Tegyük még fel, hogy az S_1 -beli kulcsok mind kisebbek az S_2 -belieknél. Javasoljunk hatékony algoritmust a két fa egyesítésére. A cél tehát egy olyan kiegészített 2-3-fa, amelyben a kulcsok $S_1 \cup S_2$ elemei.
5. Egy kezdetben üres 2-3-fába az $1, 2, \dots, n$ számokat szúrtuk be ebben a sorrendben. Bizonyítsa be, hogy a keletkezett fában a három gyerekkel rendelkező csúcsok száma $O(\log n)$.
6. Egy B_{20} -fának 10^9 levele van. Mekkora a fa szintjeinek minimális, illetve maximális száma?
7. Egy 1 000 000 rekordból álló adatállományt B-fában szeretnénk tárolni. A rekordok hossza 200 byte, a kulcs hossza 40 byte, egy mutató helyigénye 5 byte. A lapméret 2000 byte. A lapelérések számának csökkentése érdekében azt szeretnénk, hogy a B-fa minden csúcsa ráférjen egy-egy lapra. Célunk viszont, hogy emellett a lehető legjobban kihasználjuk a rendelkezésre álló helyet, így például egy levélnek megfelelő lapon több rekordot is tárolhatunk. Adjunk minél pontosabb felső becslést a szükséges lapok számára!

Gyakorlás:

1. Hogyan változhat egy piros-fekete-fában található piros csúcsok száma egyetlen elem beszúrásakor?
2. 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!
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. Tervezzon adatstruktúrát a következő feltételekkel. Természetes számokat kell tárolni, egy szám többször is szerepelhet. A szükséges műveletek:
BESZÚR(i): i egy újabb példányát tároljuk
TÖRÖL(i): i egy példányát töröljük
MINDTÖRÖL(i): i összes példányát töröljük
DARAB(i): visszaadja, hogy hány példány van i -ből
ELEM(K): megmondja, a nagyság szerinti rendezésben a K -adik elem értékét.
Az adatstruktúra legyen olyan, hogy ha m -féle elemet tárolunk, akkor mindegyik művelet lépésigénye $O(\log m)$.
5. Adott egy $n = 2^k - 1$ pontú teljes bináris keresőfa. A fában tárolt elemek egészek az $I = [1, 2^k]$ intervallumból és egy szám legfeljebb egyszer fordul elő a fában. Utóbbi feltétel szerint pontosan egy olyan $i \in I$ egész van, amely nincs a fában. Adjunk hatékony módszert i meghatározására.