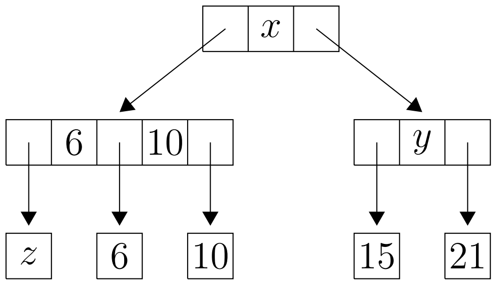
2-3 fák, hash-elés

Algoritmuselmélet

2019. tavasz

13. gyakorlat

1. Határozzuk meg, hogy milyen értékek állhatnak x, y és z helyén az alábbi, különböző pozitív egészeket tároló $2-3$ fában!



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. Egy különböző egész számokat tároló 2-3 fa gyökerében két útjelző van, a 101 és a 117. Legfeljebb hány elemet tárolhat a fa?

4. 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?

5. Adott n darab intervallum $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)$; az intervallumok végpontjai racionális számok és szokás szerint $a_i < b_i$. Tegyük fel, hogy egyik intervallum sem tartalmazza teljes egészében a másikat. Az intervallumokat a kezdő a_i koordinátájuk szerint szervezett 2-3 fában tároljuk. Hogyan lehet ennek segítségével egy adott x pontra $O(\log n)$ lépésben meghatározni, hogy a megadottak között van-e olyan intervallum, ami az x pontot tartalmazza?

6. Nyitott címzéssel hash-elünk egy kezdetben üres $M = 11$ méretű táblába a $h(x) = x \pmod M$ hash-függvénnyel lineáris próbával. Mi lesz a tábla állapota az egyes lépések után, ha a 11, 9, 99, 7, 18 külcsokat ebben a sorrendben beszúrjuk, majd töröljük a 99-et és végül beszúrjuk a 33-at?

7. A hash-függvény legyen $f(x) = x \pmod M$ és a táblaméret $M = 7$. Helyezzük el a táblában a 9, 4, 17, 13, 3 kulcsokat ebben a sorrendben kvadratikus maradék próbálást használva az ütközések feloldására, majd töröljük a 17 kulcsot. Hol ér véget a KERES(10) művelet?

8. Egy 7 méretű hash-táblába a $h(x) = x \pmod 7$ hash-függvénnyel szúrunk be elemeket. Az ütközéseket kettős hash-eléssel oldjuk fel a $h'(x) = 5 - \big(x \pmod 5\big)$ másodlagos hash-függvény segítségével. A táblába a 19, 26, 38, 33, 31 elemeket szúrjuk be ebben a sorrendben. Adjuk meg a hash-tábla állapotát minden beszúrás után!

9. Egy $M = 1000$ méretű tömbbe nyitott címzésű hash-elést végzünk a $h(x) = x \pmod{1000}$ hash-függvény segítségével. Valaki azt javasolta, hogy a lineáris próba helyett használjuk a $h_i(x) = i \cdot x \pmod{1000}$ ugrósorozatot ($i = 0, 1, \ldots, 999$). Jó próbasorozatot kapunk-e így?

10. A vödrös hash-elésnél az egyes vödrök tartalmát lapok egy-egy láncolt listájában tároljuk. Legyen a vödörkatalógus mérete M és a hash-táblában tárolt lapok összes száma L.

    (a) Legrosszabb esetben nagyságrendileg hány lépést kell tennünk egy keresés során?

    (b) Legrosszabb esetben nagyságrendileg hány lépést kell tennünk egy beszúrás során?

    Most tároljuk az egyes vödrök tartalmát láncolt lista helyett egy-egy 2-3 fában.

    (c) Legrosszabb esetben nagyságrendileg hány lépést kell tennünk egy keresés során?

    (d) Legrosszabb esetben nagyságrendileg hány lépést kell tennünk egy beszúrás során?

11. 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 ($0 \le i < n$). Legfeljebb hány ütközés lehetett, ha az ütközések feloldására

    (a) lineáris próbálást,

    (b) kvadratikus maradék próbálást használtunk?