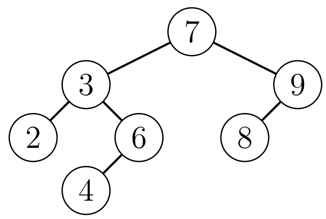
Algoritmusok és gráfok
HATODIK GYAKORLAT, 2019. október 18.
MEGOLDÁSOK néhány feladatra

1. (a) Építsen beszúrásokkal bináris keresőfát az alábbi sorrendben érkező számokból: 7,3,2,9,8,6,4.
(b) AVL-fa-e ez a fa?
(c) Szúrja be az (a) résznél kapott fába az 5-t, majd törölje ki a fából a 2, a 6 és a 7 kulcsokat és minden lépés után döntse el, hogy a kapott fa AVL-fa-e.

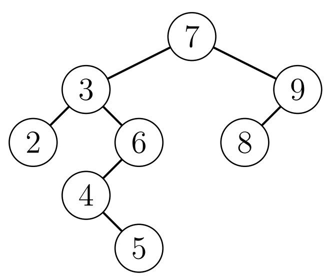
Megoldás

(a)



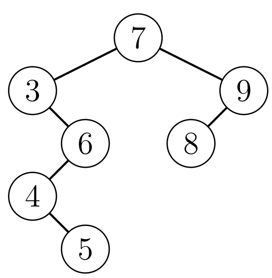
(b) Ez AVL-fa, mert minden csúcsra teljesül, hogy a jobb és a bal részfa magasságának különbsége legfeljebb 1.

(c) Az 5 beszúrása után:



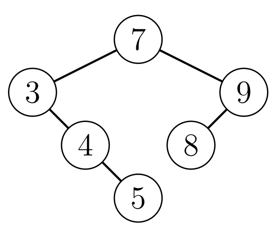
Ez már nem AVL-fa, mert a 3 jobb fájában 3 szint van, a balban meg csak 1.

A 2 törlése után:

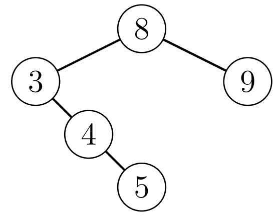


Ez pláne nem AVL-fa, mert a 3 jobb fájában 3 szint van, a balban meg nulla.

A 6 törlése után:



Ez sem AVL-fa, mert a 3 jobb fájában 2 szint van, a balban meg nulla.

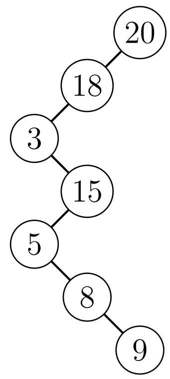


Ez sem AVL-fa, mert a 3 jobb fájában 2 szint van, a balban meg nulla.

2. 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? Ha nem lehetséges, indokolja meg miért nem, ha pedig lehetséges, határozza meg az összes olyan x egész számot, amire ez megtörténhet.

   Megoldás

   Lehetséges, például az alábbi bináris keresőfában:

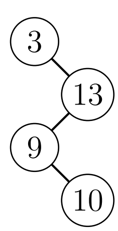
   

   Ez a fa bináris keresőfa, mert minden csúcstól jobbra csak nála nagyobbak, balra pedig csak nála kisebbek vannak. Ha a keresés során ezeket a számokat látjuk ebben a sorrendben, akkor a keresett szám vagy a 9-es (és akkor ez egy sikeres keresés) vagy a 9-nél akadunk el, ekkro pedig a keresett számra igaz, hogy 20-nál kisebb (mert a 20-ból balra mentünk), 18-nál is kisebb, de 3-nál nagyobb, 15-nél kisebb, 5-nél nagyobb és 8-nál nagyobb. Ezeket a feltételeket összegezve a szám nagyobb, mint 8, de kisebb, mint 15, vagyis 10, 11, 12, 13 vagy 14.

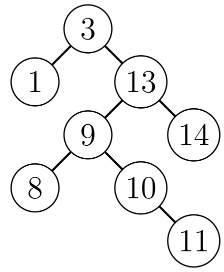
3. **(ZH 2018)** Egy bináris keresőfában az $1, 3, 8, 9, 10, 11, 13, 14$ számokat tároljuk valamilyen elrendezésben és tudjuk, hogy amikor a 10-et keressük, akkor a keresés során először a 3-as számot látjuk, utána a 13-t, majd a 9-et, végül pedig a 10-et. Rajzolja fel azt a 8 csúcsú bináris keresőfát, ahol ez megtörténhetett, majd lássa be, hogy a fa csak így nézhet ki.

   Megoldás

   A keresési útvonalból következik, hogy a fában a $3, 13, 9, 10$ csúcsok így helyezkednek el:

   Az a kérdés csak, hogy a többi tárolt elem hol lehet ezekhez képest. Az 1 az egyetlen 3-nál kisebb elem, vagyis a 3 bal fájában csak ő van, a 8-as az egyetlen elem, ami 3-nál nagyobb, de 9-nél kisebb vagyis ő lesz a 9 bal gyereke, a 11 az egyetlen olyan elem, ami 13-nál kisebb, de 10-nél nagyobb vagyis ő lesz a 10 jobb gyereke és 14 az egyetlen 13-nál nagyobb szám, ő lesz 14 jobb gyereke. Tehát a fa csak így nézhet ki:



4. **(PPZH 2018)** Egy bináris keresőfában csupa különböző egész számot tárolunk. Lehetséges-e, hogy egy x érték keresése során az alábbi számokat járjuk be ebben a sorrendben: $3, 10, 8, 9, 12, 5$? Ha úgy véli, hogy lehetséges, akkor mutasson egy olyan bináris keresőfát és olyan x értéket, ahol ez megtörténhetett, különben pedig magyarázza meg, hogy miért nem lehetséges ez.

   Megoldás

   Ha a keresés során a 10 után a 8-at látjuk, akkor a keresett szám kisebb, mint 10, ezért megyünk a 10 bal fájába. Ebben a fában azonban nem láthatunk már 10-nél nagyobb értékeket, tehát pl. a 12-t nem láthatjuk vagyis ez a keresési út nem lehetséges.

5. **(Vizsga 2018)** Adott két teljes bináris keresőfa, mindegyikben n elemet tárolunk. Adjon $O(\log n)$ elemszámú eljárást, ami eldönti, hogy igaz-e, hogy az első fa minden eleme nagyobb, mint a második fa minden eleme.
   (Emlékeztetőül: a teljes bináris fa olyan bináris fa, amiben minden szint tele van.)

   Megoldás

   Algoritmus: Keressük meg az első fa legkisebb és a második fa legnagyobb elemét a tanult eljárással. Ha az első fa legkisebb eleme nagyobb, mint a második fa legnagyobb eleme, akkor az első fa minden eleme nagyobb, mint a második fa minden eleme, egyébként pedig az első fának van olyan eleme (pl. a minimum), ami nem nagyobb a másik fa egy eleménél (pl. a maximummnál).

   Jóság: Ha az első minimuma nagyobb, mint a második maximuma, akkor az első minden eleme nagyobb, mint a második maximuma vagyis nagyobb, mint bármely elem a második fában. Egyébként pedig találtunk egy olyan párt (első minimuma és második maximuma), ami mutatja, hogy nem teljesül a kívánt állítás.

   Lépésszám: A minimum és a maximum keresése is a szintszámmal arányos, de egy teljes bináris fában a szintek száma $O(\log n)$ vagyis ezek megvannak $O(\log n)$ lépésben, utána meg már csak egy összehasonlítás kell.

8. Nyílt címzéssel hasheltünk egy 11 elemű táblába a $h(k) = k$ maradéka 11-gyel osztva hash-függvény és lineáris próbát használva.
   (a) A következő kulcsok érkeztek (ebben a sorrendben): $6, 5, 11, 17, 16, 3, 2, 14$. Hogyan néz ki a tábla végső állapota?
   (b) Hogyan zajlik az a) pontban kapott táblában a keres(16) művelet? És a keres(4)?
   (c) Törölje az (a) pont táblájából a 2-es számot, majd keresse a 15-öt a táblában. Mi történik, hol ér véget a keresés?
   (d) Szúrja be most a 25-öt a táblába. Hova kerül végül a 25-ös szám?

   Megoldás

   (a)

0	1	2	3	4	5	6	7	8	9	10
11	2	3	16	17	5	6				14

   (b) A keres(16) esetén kiszámoljuk a 16-hoz tartozó hash értéket, ami 5, majd az 5-ös indexű cellából elindulva balra addig megyünk, míg a 16-ot meg nem találjuk. A keres(4) esetén a 4-es indexű cellában kezdünk keresni (mert h(4) = 4) és innan megyünk balra, amíg a 9-es indexnél üres cellát látunk, tehát az elem nincs a tömbben.

0	1	2	3	4	5	6	7	8	9	10
11	*	3	16	17	5	6				14

Ha ebben keressük 15-öt, akkor $h(15) = 4$ miatt a 4-es indexű cellából indulva megyünk balra addig, amíg üreset nem találunk (a törölt cellánál nem állunk meg!), a keresés a 9-es indexnél ér véget.

(d) A 25 beszúrását a $h(25) = 3$-as cellában kezdjük, innen megyünk balra, amíg törölt vagy üres cellát nem találunk, ez az 1-es indexnél lesz, ide fogjuk beszúrni az elemet.