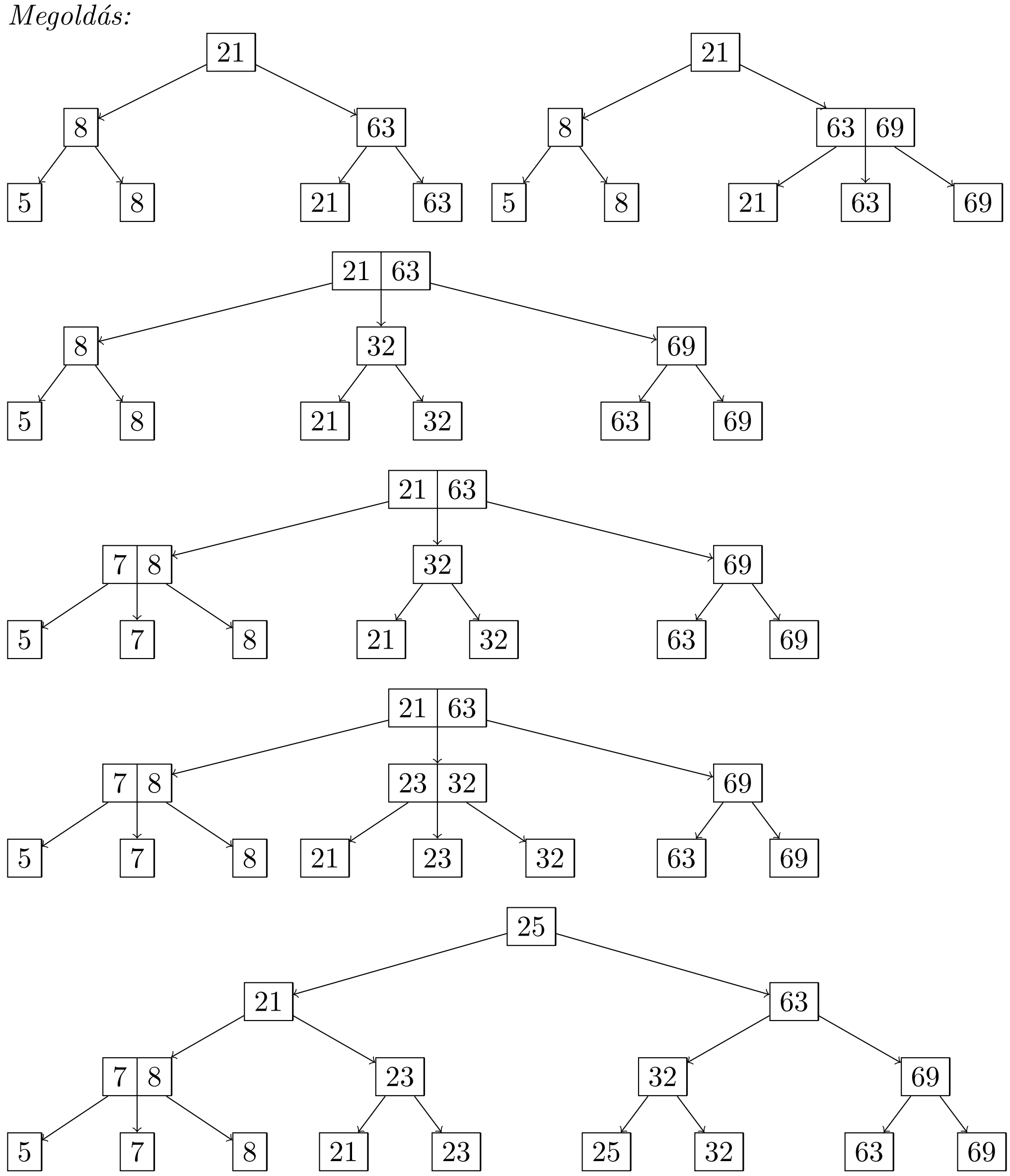
2-3-fa, hash

1. Egy 2-3-fában 4 elem van. Egyértelmű-e a 2-3-fa?

   Megoldás: A szintek száma $\log_3 4+1 = 2.2618\ldots$ és $\log_2 4+1 = 3$ között van, tehát pont 3. A három szintű fának pont 4 levele van, ha minden csúcsnak pont 2 gyereke van. Ha bármelyiknek 3 gyereke lenne, már legalább 5 levél lenne. Tehát a fa egy teljes bináris fa, ami egyértelmű. Mivel azt is tudjuk, hogy az elemek a levelekben balról jobbra növekednek, valamint a belső csúcsok címkéi is egyértelműen meghatározhatóak a levelekből, ezért a 2-3-fa egyértelmű.

2. Adjon egy 2-3-fát amely az 5,8,21,63 elemeket tartalmazza, majd sorban szúrja be a 69,32, 7,23,25 elemeket!

   Megoldás:



3. Egy 2-3-fa gyökerének három fia van, a benne szereplő két érték 40 és 50. Mennyi lehet a tárolt elemek minimális, illetve maximális száma, ha tudjuk, hogy csak pozitív egész számokat tárol a fa?

   Megoldás: A feltételekből következik, hogy a gyökérnek 3 fia van, a balban 1 és 39 között, középen 40 és 49 között, a jobb oldalon 50-től tárolunk elemeket.

   A minimális elemszámot nyilván úgy kapjuk, ha a gyökér gyerekei levelek, az x, 40, 50 elemeket tárolja a fa, ahol $1 \leq x \leq 39$ tetszőleges egész.

   A maximális elemszámhoz nézzük, mit tudunk mondani a fa magasságáról! Mivel jelen esetben a három részfa közül a középső tárolhatja a legkevesebb elemet, ez fogja a korlátot adni. Ennek a magassága legfeljebb

3, és a lehetséges maximális 10 elem tárolásához kell is ennyi. Tehát a másik két részfa magassága is 3 lesz. Ezekben akkor kapjuk a lehető legtöbb levelet (tárolt elemet), ha mindig 3 felé ágaznak, azaz 27 − 27 elemet tárolnak. Ennyi megengedett elem van is mindkét oldalon, tehát a maximális elemszám $10 + 2 \cdot 27 = 64$.

4. Az $[1, 178]$ intervallumba eső összes egész számot egy 2-3-fában tároljuk. Tudjuk, hogy a gyökérben két útjelző van, és az első ezekből a 17. Mi lehet a második?

   Megoldás: A második útjelző a jobb oldali részfában tárolt legkisebb szám, tehát ezt kell meghatároznunk.

   Az első útjelző miatt a bal részfa 1-től 16-ig tárolja a számokat. Ezért ennek a részfának a magassága legalább 3 és legfeljebb 4. Mivel két útjelzőnk van, a gyökérnek 3 gyereke van. Tudjuk, hogy mindhárom részfa magassága ugyanaz. Ha ez a közös magasság 3, akkor a középső és jobb részfában is legfeljebb $3^3 = 27$ levél van, tehát legfeljebb ennyi értéket tárolhat, így nem fér a fába mind a 178 elem. Tehát a bal részfa, és így a másik kettő magassága is 4 kell legyen. Ekkor a másik két részfa mindegyike legfeljebb $3^4 = 81$ értéket tárolhat. Mivel $16 + 2 \cdot 81 = 178$, ezért csak az lehet, hogy a középső és jobb részfában minden nem levél csúcsnak 3 gyereke van (a bal részfában meg mindenhol 2), a középső és jobb részfa is 81 elemet tárol.

   A gyökérben levő második útjelző a jobb részfa legkisebb eleme, ami ezek szerint $16 + 81 + 1 = 98$.

5. Nyitott címzéssel hash-elünk egy kezdetben üres $M = 11$ méretű táblába a $h(x) = x \pmod M$ hash-függvénnyel. Mi lesz a tábla állapota, ha a 4, 5, 14, 15, 16, 26, 3 kulcsokat a megadott sorrendben beszúrjuk és az ütközések feloldására

   (a) lineáris próbát használunk?

   (b) kvadratikus maradék próbát használunk?

   (c) kettős hash-elést használunk, amikor $h'(x) = 7x \pmod{(M-1)}$ a második hash-függvény?

   Hány ütközés történt az egyes esetekben?

   Megoldás: (a) Az ütközések száma: $0 + 0 + 0 + 2 + 4 + 4 + 4 = 14$, a végén a tábla elemei sorban 22, 16, 15, 14, 4, 5, –, –, –, –, 3

   (b) Az ütközések száma (lefelé indulok): $0 + 0 + 0 + 3 + 2 + 4 + 1 = 10$, a végén a tábla elemei sorban 15, –, 3, 14, 4, 5, 16, –, 26, –, –

   (c) Az ütközések száma (most is lefelé indulok): $0 + 0 + 0 + 1 + 2 + 1 + 3 = 7$, a végén a tábla elemei sorban 3, 16, 26, 14, 4, 5, –, –, –, –, 15

6. Előfordulhat-e nyitott címzéses hash-elés esetén, hogy az $n > 3$ méretű táblában csak 3 elem van, de a keresés lépésszáma n ?

   Megoldás: Igen, ha korábban törlések is történtek, pontosabban, ha minden helyről, ahol most nincs elem volt törlés. Ekkor a keresés ezeken továbbmegy, és ha a 3 bent levő egyike sem a keresett elem, akkor mindenképpen bejárjuk az egész táblát.

7. Jó választás-e $M = 7$ méretű táblánál az $h(x) = x^2 \pmod 7$ hash-függvény?

   Megoldás: Nem, mert $f(1) = f(6)$, $f(2) = f(5)$ és $f(3) = f(4)$, tehát minden a 0, 1, 4, 2 helyek egyikénél kezd ahelyett, hogy a 7 lehetséges helyre szétoszlanának (több ütközés várható).

8. A $T[0 : M]$ táblában $2n$ elemet $(n < M/3)$ helyeztünk el valamilyen hash-függvény segítségével, amikor azt tapasztaltuk, hogy az elemek mindegyike az első $3n$ hely egyikére került. Ha nem volt közben törlés és a végén a táblában minden $3i$ indexű hely üres maradt $(0 \leq i < n)$, akkor legfeljebb hány ütközés lehetett, ha

   (a) lineáris próbát használtunk?

   (b) kvadratikus maradék próbát használtunk?

   Megoldás: (a) Ezek szerint a táblázat elején mindig két foglalt helyet követ egy üres. Lineáris próbánál csak a két szomszédos ütközhetett a másodiknak érkező beszúrásakor, tehát legfeljebb n ütközés volt. Ennyi lehet is, pl. ha az elemek sorban 2, $M + 2$, 5, $M + 5$, stb.

   (b) Mivel mindenkinek legalább az egyik szomszédja üres, elemenként megint legfeljebb egy ütközés lehetett. Ezért az ütközések csak egy-egy szomszédos páron belül történhettek. Azaz most is legfeljebb n ütközés lehetett. Az előző példa mutatja, hogy ez lehetséges is.

9. Egy m méretű hash-táblában már van néhány elem. Adjon $O(m)$ lépésszámú algoritmust, amely meghatározza, hogy egy újabb elem lineáris próbával történő beszúrásakor maximum hány ütközés történhet!

Megoldás: A feladat az, hogy a táblában levő legnagyobb foglalt blokk hosszát meghatározzuk. Ezt egyszerűen dinamikus programozással lineáris időben megtehetjük például így: Legyen M az eddigi legnagyobb befejezett blokk hossza, B pedig az aktuális érték, kezdetben mindkettő 0. Az m méretű T táblában sorban az $i = 0,1,2\ldots$ indexekre (az indexet $\mod m$ értve):
Ha $T[i]$-ben nincs elem és $B > M$, akkor legyen $M = B$.
Ha $T[i]$-ben nincs elem, akkor $B = 0$.
Ha $T[i]$-ben van elem, akkor növeljük B-t eggyel.
Ha $i \geq m$ és $T[i]$-ben nincs elem, akkor megállunk. (Ez nyilván be fog következni valamely $i < 2m$ esetben.)
A végén $\max(M, B)$ a válasz.

10. A $b_0 ... b_n$ alakú $n+1$ hosszú bitsorozatokat akarjuk tárolni. Tudjuk, hogy a b_0 paritásbit (ami a sorozatban az egyesek számát párosra egészíti ki). Ha nyitott címzésű hash-elést használunk $h(x) \equiv x \pmod M$ hash-függvénnyel és lineáris próbával, akkor $M = 2^n$ vagy $M = 2^n + 1$ méretű hash-tábla esetén lesz kevesebb ütközés?

Megoldás: Az ilyen számokon a mod 2^n levágja az első bitet, tehát $M = 2^n$ esetén különböző sorozatok nem ütköznek. Az $M = 2^n + 1$ esetben viszont lesz ütközés, a $00 \cdots 00$ ütközik az $10 \cdots 01$ sorozattal, hiszen mindkét esetben a h értéke 0.