Az alábbi pszeudokód inputja egy $n \geq 9$ szám és egy n méretű $A[0 : n-1]$ tömb, melyben csupa pozitív egész számot tárolunk. A pszeudokódban egy lépésnek az értékadás és az összeadás számít. Igaz-e, hogy ennek a kódnak a lépésszáma $O(n)$?

```
i := 1
j := n-1
ciklus amíg i < j:
    A[i] : = A[i] + 17
    i := i + 1
    j := j - 1
ciklus vége

ciklus k = 0 -tól 7-ig:
    ciklus s = (n -1)-tól (n - 8)-ig:
        A[s] := A[s] + A[k] + 3
    ciklus vége
ciklus vége
```

$O(1)$

$O(n)$

$O(1)$

$O(1)$

lefut hányszor?

$i = 1$ $j = n-1$ => különbségük $n-2$

és $\forall$ futáskor csökken a különbség 2-vel

=> $\leq \frac{n}{2}$ futás után $i \geq j$ =>

lefut $O(n)$-szer és egy futás $O(1)$ =>

ez a ciklus $O(n)$

lefut 8-szor => $O(1)$

lefut 8-szor fut le → egész 2. rész $O(1)$

teljes LSZ: $O(1) + O(n) + O(1) \Rightarrow O(n)$ ✓

A 2, 3, 8, 5, 1, 7, 4 tömböt rendezzük. Eközben előállhat-e az 1, 2, 3, 5, 8, 7, 4 sorrend, ha
(a) beszúrásos rendezést használunk?
(b) kiválasztásos rendezést használunk?

a)

2 | 3, 8, 5, 1, 7, 4

2, 3 | 8, 5, 1, 7, 4

2, 3, 8 | 5, 1, 7, 4

2, 3, 5, 8 | 1, 7, 4

1, 2, 3, 5, 8 | 7, 4 => pont kijött ✓

rendezett rész | köv. elem .....

balra cserélgetem a helyére érünk a rendezett részben

b)

2, 3, 8, 5, 1, 7, 4

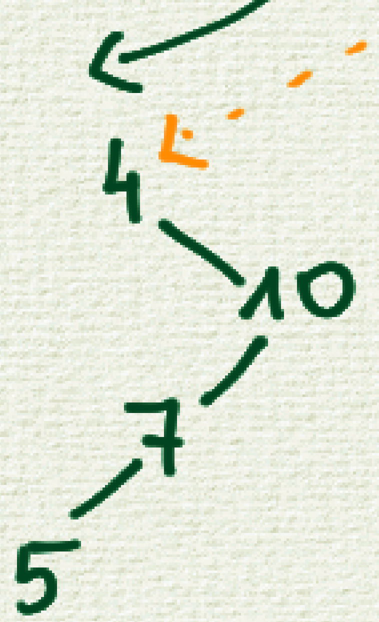
1, 3, 8, 5, 2, 7, 4

1, 2, 8, 5, 3, 7, 4

1, 2, 3, 5, 8, 7, 4 => kijött ✓

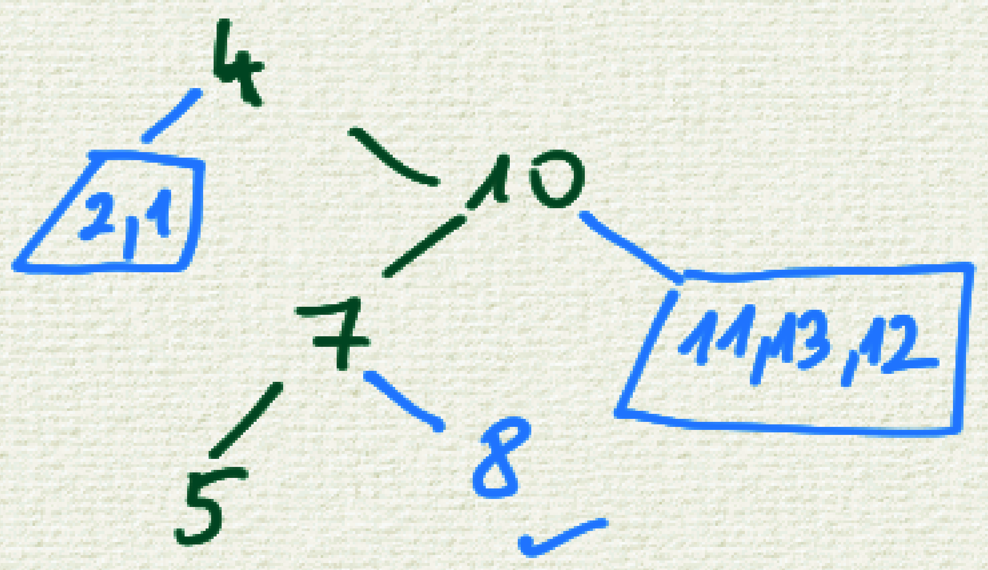
Egy bináris keresőfában az 5-ös szám keresésekor az alábbi számokat látjuk ebben a sorrendben addig, amíg a keresett 5-ös értéket megleljük: 4, 10, 7, 5. Tudjuk továbbá, hogy ezt a fát posztorder bejárással bejárva a csúcsokat 2, 1, 5, 8, 7, 11, 13, 12, 10, 4 sorrendben látogatjuk meg. Adja meg azt az egyetlen bináris keresőfát, amiben ez előfordulhat és mutassa meg, hogy csak ez a fa lehetséges.

(1) Ez benne van a fában:



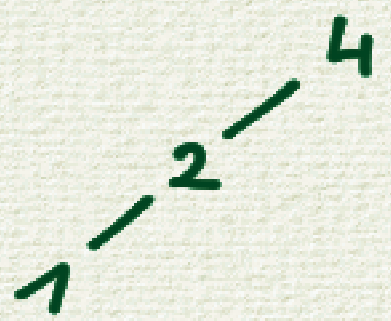
gyökérnél kezdik keresni és mindig gyerekbe megyek a keresés során + bin kerfa tul miatt 10 a 4-től jobbra......

(2)

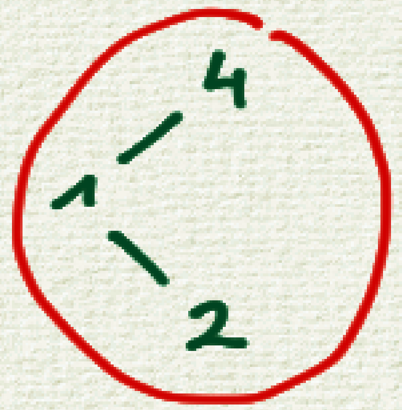


postorder => ez az összes csúcs és binkerfa tul miatt csak itt lehetnek ezek az értékek

(3) ? 2, 1 hogy áll egymáshoz képest



vagy

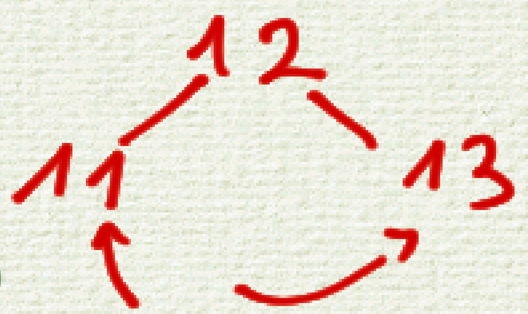


postorderben 2 az 1 előtt van <=> 1 jön később => 1 a szülője a 2-nek

(4) 11, 12, 13 hogy áll?

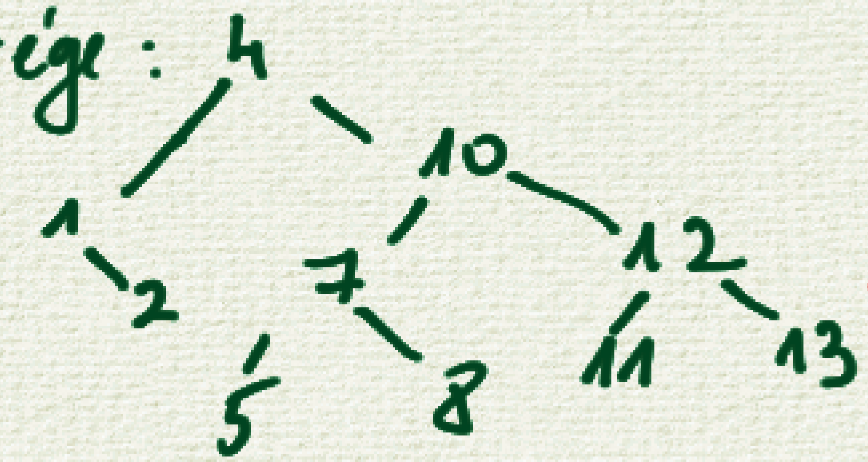
postorder: 11, 13, 12

ő a gyökér



bin. kerfa miatt

vége:



1

Egy 11 méretű hash táblába 7 kulcsot szúrtunk be nyílt címzéssel, lineáris próbával, a használt hash függvény a $h(K) =$ K maradéka 11-gyel osztva függvény volt. (A lineáris próba lefele indul.) Törlés nem történt.

Ezután a hash táblát lemásoltuk kézzel egy papírra, de sajnos az egyik számot rosszul írtuk le.

(a) Melyik a rosszul leírt szám, ha ez a lemásolt tábla?

0	1	2	3	4	5	6	7	8	9	10
1	12		14	5	6	17				16

nem lehet ez itt

16

(b) Mi lehetett az az x érték, amit rosszul másoltunk le, ha tudjuk, hogy $20 \leq x \leq 30$ és x máshol nem szerepelt a táblában?

a) 16 mert $h(16) = 5$ => 5-ös cellába került volna, de ha az foglalt volt => balra indult.
Legkésőbb 2-be bekerült volna, hisz az üres

b) Mi lehetett a 16 helyett?

$h(x) = 10$ lehetett → 21
vagy
x lecsúszott 0-s vagy 1-es cellából, azaz?

$h(x) = 0$ → 22

$h(x) = 1$ → 23

és máshol nem lehetett, mert lefele menve balra legkésőbb 2-be került volna az x.

Adott egy $n \geq 2$ méretű tömb, melyben csupa különböző egész számot tárolunk. Adjon $O(n \log n)$ lépésszámú algoritmust, ami eldönti, hogy igaz-e, hogy a tömbben szereplő bármelyik számnak vagy a fele vagy a kétszere is benne van a tömbben.

ötlet: meg kéne nézni $\forall$ $A[i]$ számra, hogy $2 \cdot A[i]$ vagy $\frac{A[i]}{2}$ $\overset{?}{\in}$ tömb

→ ezt kéne ügyesen

Algo: 1) rendezzük a tömböt összefésüléses rendezéssel → A tömb $O(n \cdot \log n)$

2) $\forall$ $i = 0$-tól $(n-1)$-ig] n-szer fut le

$2 \cdot A[i]$-t és $\frac{A[i]}{2}$-t keresem a tömbben bináris kereséssel → 2db bin. keresés $= 2 \cdot O(\log n) \Rightarrow O(\log n)$

és ha van olyan i, hogy se $2 \cdot A[i]$, se $\frac{A[i]}{2}$ $\notin$ tömb $\Rightarrow$ "Nem"

ha $\forall$ i-re vagy $A[i]$ felét vagy kétszeresét megtalálom

Eben: "Igen"

Jóság: $\forall$ elemet megnézek, hogy fele/kétszerese $\overset{?}{\in}$ tömb és használható a bin. keresés, mert 1) után a tömb rendezett

LSZ: 2) pont $n \cdot O(\log n) \Rightarrow O(n \cdot \log n)$ egész: $O(n \cdot \log n) + O(n \cdot \log n) \Rightarrow O(n \cdot \log n)$

Egy szomszédossági mátrixával adott $n > 19$ csúcsú, irányított G gráfban 17 csúcs pirosra van színezve, a többi csúcs színtelen. A színezés egy n méretű, a csúcsokkal indexelt C tömbben adott (ha a csúcs piros, akkor $C[v]$ = piros, különben $C[v]$ = színtelen).
Adott továbbá két kijelölt színtelen csúcs, A és B és A-ból szeretnénk B-be eljutni a gráf éleit használva úgy, hogy legfeljebb egy piros csúcson megyünk át. Adjon $O(n^2)$ lépésszámú algoritmust, ami a szomszédossági mátrix különböző módosításaival és egy tanult algoritmus változtatás nélküli (többszöri) futtatásával eldönti, hogy ez lehetséges-e és ha lehetséges, akkor azt is megadja, hogy hány élből áll egy, a lehető legkevesebb élet használó ilyen eljutás.

Egy szomszédossági mátrixával adott $n > 19$ csúcsú, irányított G gráfban 17 csúcs pirosra van színezve, a többi csúcs színtelen. A színezés egy n méretű, a csúcsokkal indexelt C tömbben adott (ha a csúcs piros, akkor $C[v]$ = piros, különben $C[v]$ = színtelen).
Adott továbbá két kijelölt színtelen csúcs, A és B és A-ból szeretnénk B-be eljutni a gráf éleit használva úgy, hogy legfeljebb egy piros csúcson megyünk át. Adjon $O(n^2)$ lépésszámú algoritmust, ami a szomszédossági mátrix különböző módosításaival és egy tanult algoritmus változtatás nélküli (többszöri) futtatásával eldönti, hogy ez lehetséges-e és ha lehetséges, akkor azt is megadja, hogy hány élből áll egy, a lehető legkevesebb élet használó ilyen eljutás.

BFS kell

LSZ:

ezeknek (17 db-k) veszem a min-át

ötlet: ∀ piros csúcsra lesz 1 futás: meghatározzuk azt, hogy min. hány él kell ha ezt a pirosat érinthetjük csak

Mivel 17 csúcs van ⇒ LSZ 17 · $\boxed{}$ ⇒ uaz lesz mint 1 futás LSZ-a

→ 1 konkrét p = pirosra mit kell csinálni?

Algo: 1) [G-ből kivesszük ∀ olyan élet ami pirosba / pirosból vezet (kivéve p-ből / p-be)

↳ végig a szomszédossági mátrixon és ha $A[i,j] = 1$ (azaz $i \to j$)
és $\underbrace{i \text{ piros, de} \neq p}_{O(1)}$ vagy $\underbrace{j = \text{piros, de nem a } p}_{O(1)}$ ⇒ $A[i,j] := 0$

$O(n^2)$

n^2 cella, ∀ cella $O(1)$

jóság: 1) alatt
G'-ben út $A \rightsquigarrow B$
⇕
G-ben van út $A \rightsquigarrow B$
legfeljebb egy piroson át, p-n

2) BFS G'-ben A-ból ⇒ távolság[B] = min hány élű út van G'-ben
$A \rightsquigarrow B$ = min. él $A \rightsquigarrow B$ max egy piros van, az a p csúcs

↳ n csúcson $O(n^2)$

egész $17 \cdot (O(n^2) + O(n^2)) = 17 \cdot O(n^2) \Rightarrow O(n^2)$ ✓