$O(n)$

Az alábbi pszeudokód inputja egy $n \geq 2$ méretű $A[0 : n-1]$ tömb, melyben csupa egész számot tárolunk, negatív számok és nulla is előfordulhat. A pszeudokódban egy lépésnek az összehasonlítás, az értékadás és az összeadás számít. Igaz-e, hogy ennek a kódnak a lépésszáma $O(n)$? Válaszát indokolja!

```
ciklus i = 0-tól (n-1)-ig:
    ha A[i] < 0:
        A[i]:= 1
ciklus vége

ciklus j = 0-tól 17-ig:
    ciklus i = 0-tól (n-1)-ig:
        A[i] := A[i] + j
    ciklus vége
ciklus vége
```

→ lefut n-szer

$O(1)$ → $O(n)$ mert $n \cdot O(1)$

→ $O(1)$-szer fut le

→ lefut n-szer

$O(1)$

$\Rightarrow n \cdot O(1) \Rightarrow O(n)$

$O(n)$

$O(n)$

1. rész 2. rész

$\text{teljes} = O(n) + O(n) = O(n)$

Az $5, 2, x, 14$ tömböt összefésüléses rendezéssel rendezzük, ahol x egy olyan egész szám, ami máshol nem szerepel a tömbben. Adja meg x összes lehetséges értékét, ha
(a) x-et a rendezés során csak egyetlen másik elemmel hasonlítjuk össze.
(b) x-et a rendezés során mindhárom másik elemmel összehasonlítjuk.
Mindkét esetben magyarázza is el, hogy miért csak ezek az értékek lehetségesek.

5, 2, x, 14

2, 5   x, 14 ($x < 14$) VAGY 14, x ($14 < x$)

a) x-et csak 14-gyel hasonlítom

ha $x < 14$ lenne $\Rightarrow$ 2, 5   x, 14 $\Rightarrow$ x is 2 hasonlítva lesz

14, x lesz a sorrend $\Leftrightarrow$ $x > 14$

~~2~~, ~~5~~   14, x

2, 5, 14, x

x valóban nem hasonlítódik senki mással

b) előzőekből $\Rightarrow$ csak akkor lehet 2-vel hasonlítva, ha $\boxed{x < 14}$

~~2~~, 5   x, 14

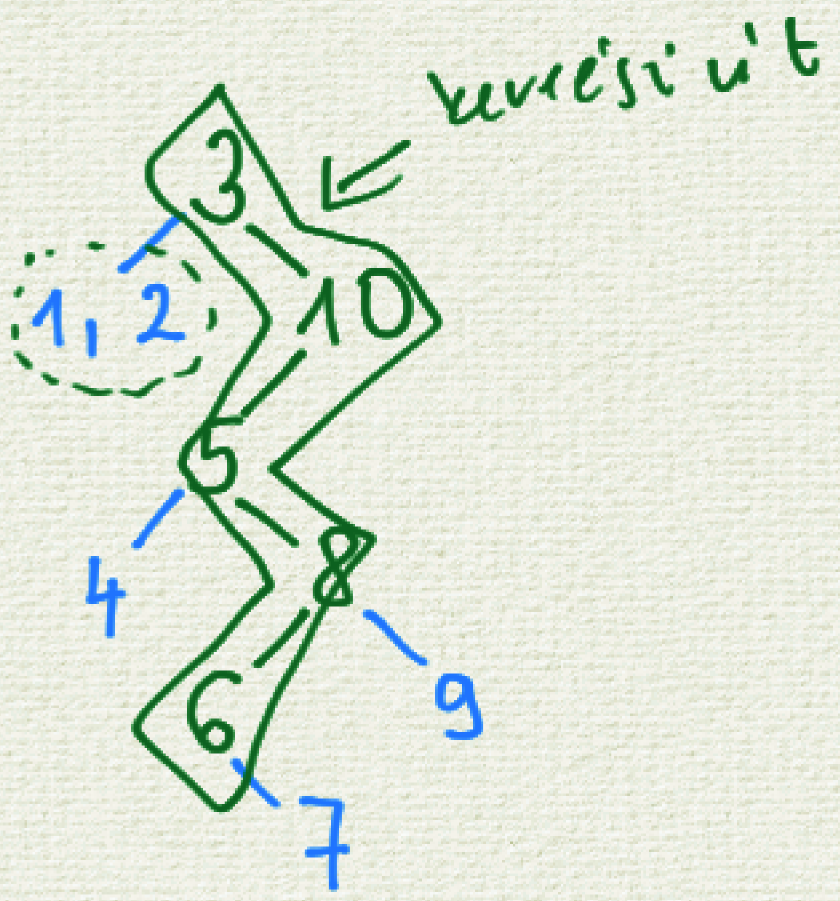
2, 5 / x

1. öh: 2 – x ✓ és ha $x < 2$ $\Rightarrow$ x nem lesz többet hasonlítva $\Rightarrow$ tehát $\boxed{x > 2}$

2. öh: x – 5 ✓

tehát $x = 3, 4, 6, 7, 8, 9, 10, 11, 12, 13$

Egy bináris keresőfában az $1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ számokat tároljuk. A 6-os szám keresésekor az alábbi számokat látjuk ebben a a sorrendben addig, amíg a keresett 6-os értéket megleljük: $\boxed{3, 10, 5, 8, 6.}$ Adja meg az összes olyan bináris keresőfát, amiben ez előfordulhat és mutassa meg, hogy csak ezek a fák lehetségesek.

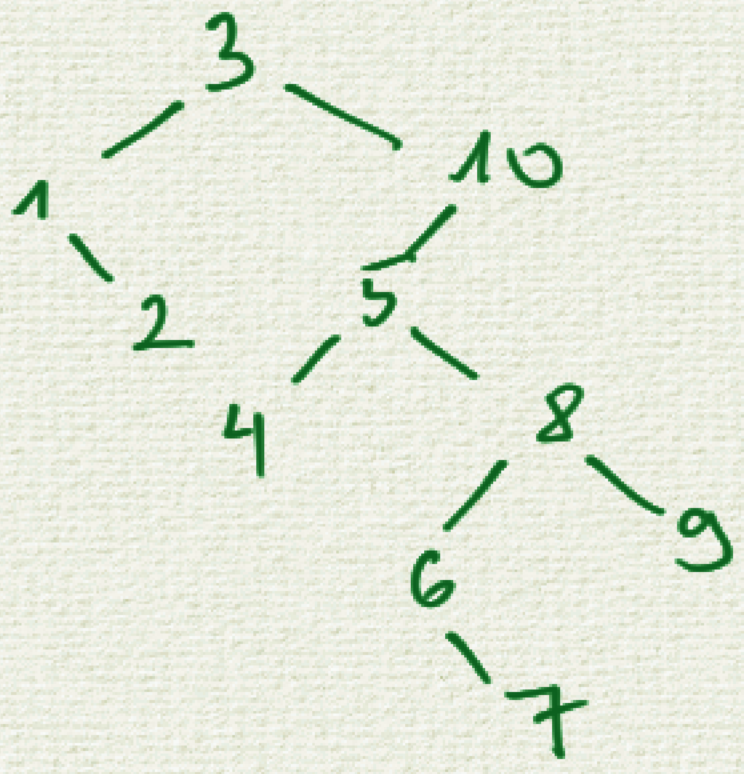


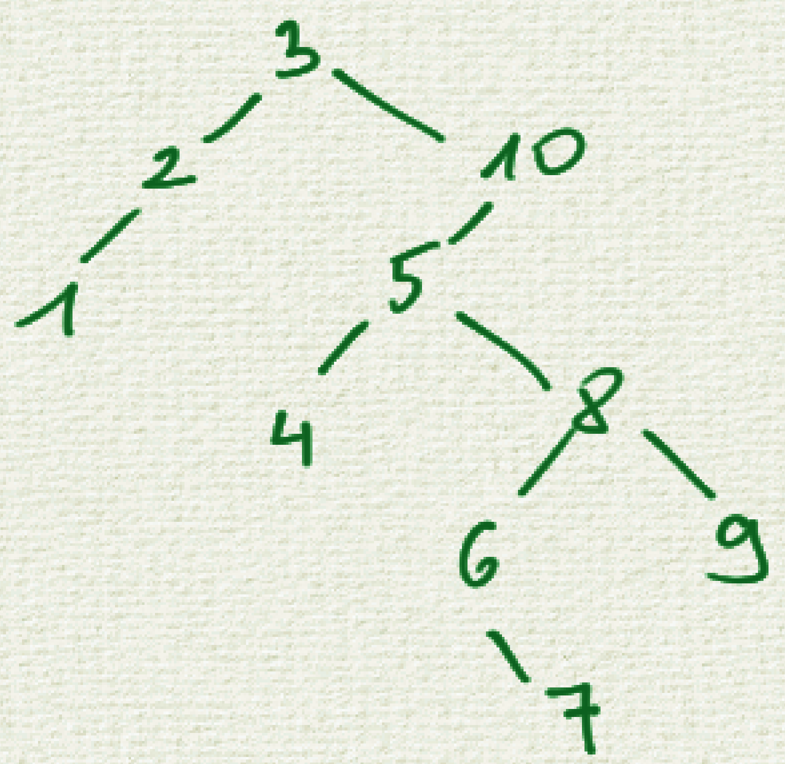
?: Hol van 1, 2, 4, 7, 9,

a bináris kerfa tul miatt

4, 7, 9 helye adott

1, 2 kétféleképp lehet





Egy nyolc csúcsú irányítatlan gráfban a BFS (szélességi bejárás) algoritmusát futtatjuk az A csúcsból úgy, hogy ha a szomszédok végigjárása során választási lehetőség adódik, akkor mindig az ábécé szerint előbb levő csúcsot választjuk. A futás végén a *honnan* tömb így néz ki:

A	B	C	D	E	F	G	H
A	C	A	C	A	G	D	E

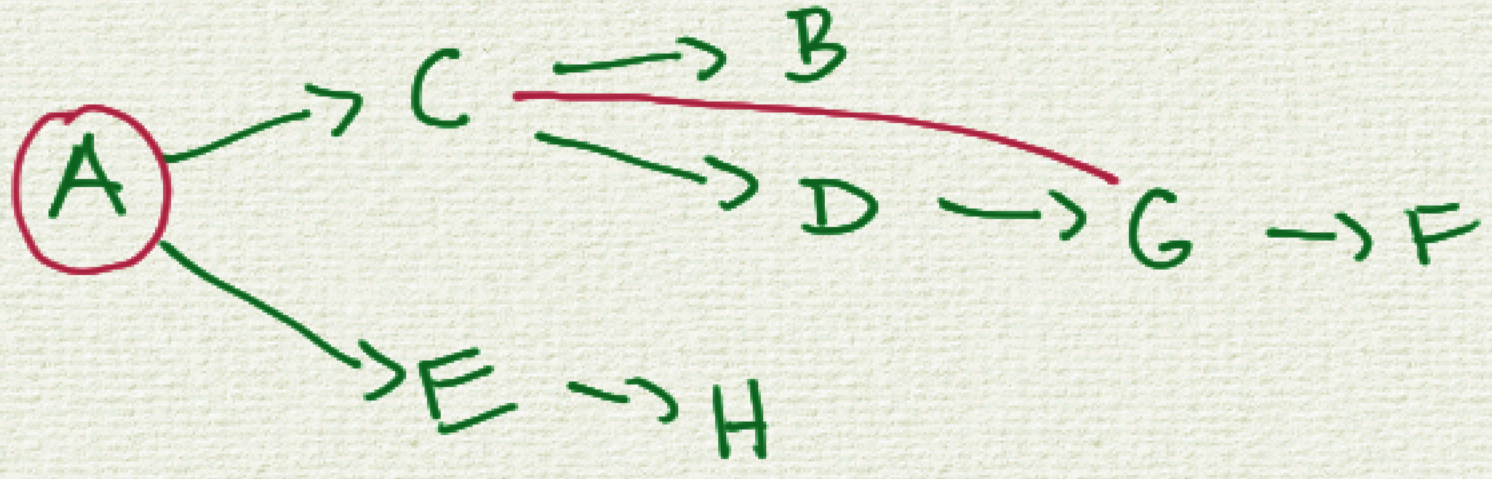
(a) Rajzolja fel a BFS fát és röviden (1-2 mondatban) írja le, hogy hogyan kapta a fát a tömbből.
(b) Ugyanezen a gráfon (melynek a BFS fa élein kívül számos más éle is lehet) futtatjuk a DFS (mélységi bejárás) eljárást szintén az A csúcsból úgy, hogy ha a szomszédok végigjárása során választási lehetőség adódik, akkor ebben az esetben is az ábécé szerint előbb levő csúcsot választjuk. Lehetséges-e, hogy a DFS eljárás a G csúcsot a C csúcsból a CG élen keresztül járja be? Válaszát indokolja!

a)

honnan[v] = v-t melyik csúcsból jártam be

a)

DFS



Nem lehet CG él a gráfban:

ha lenne, akkor BFS

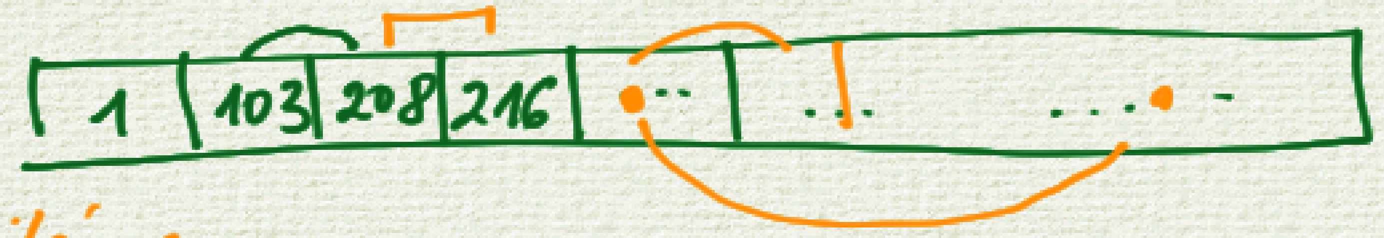
G-t C-ből járnám be a BFS-ben is

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 bármely két szám különbsége legalább 100.

∀ párt megnézünk $\binom{n}{2}$, nem $O(n \cdot \log n)$

$$\binom{n}{2} = \frac{n(n-1)}{2} \approx n^2$$

jóság



Algo

1) rendezzük őt rendezéssel $O(n \cdot \log n)$

2) végig a rendezett tömbön és ∀ szomszédos párra: $O(n)$

$$A[i+1] - A[i] \overset{?}{\geq} 100$$

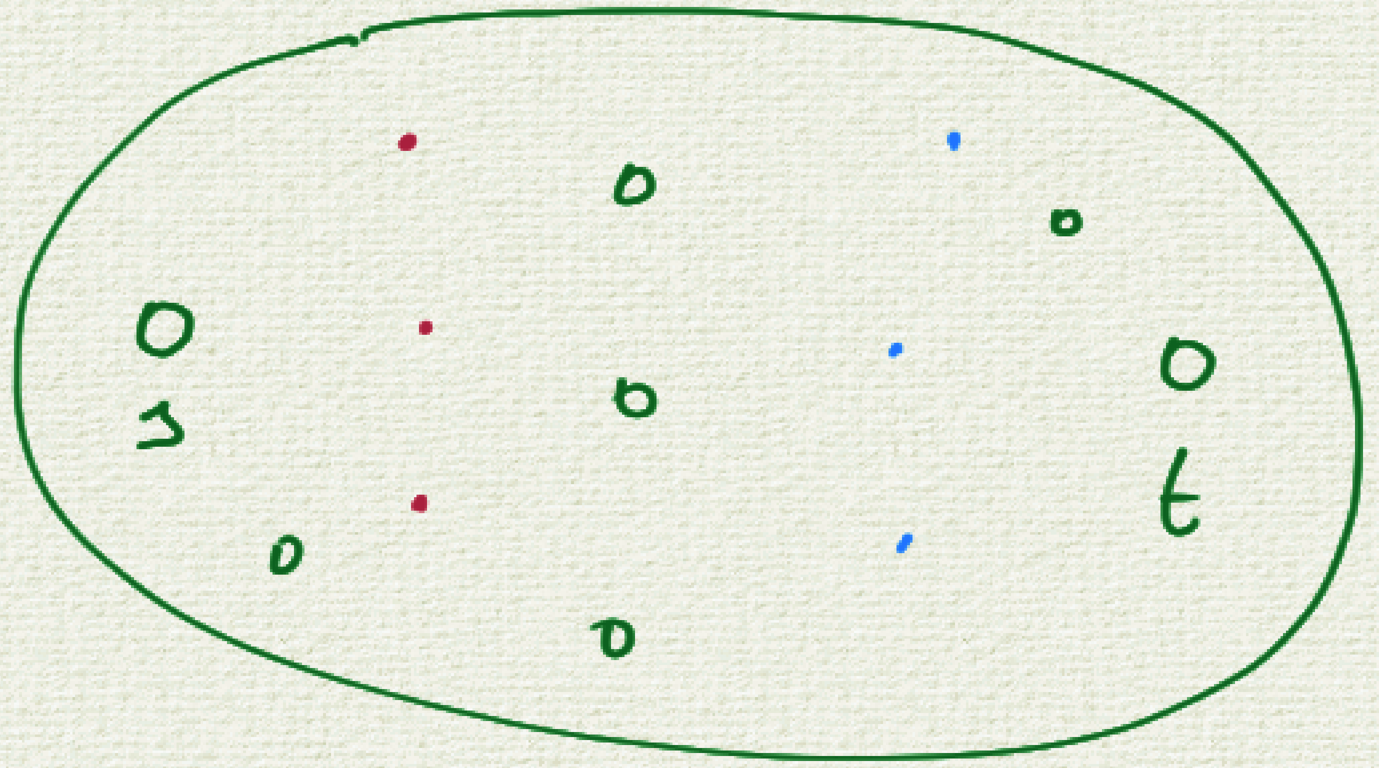
és ha van olyan pár ahol a különbség < 100 => van rossz pár, a válasz "Nem"

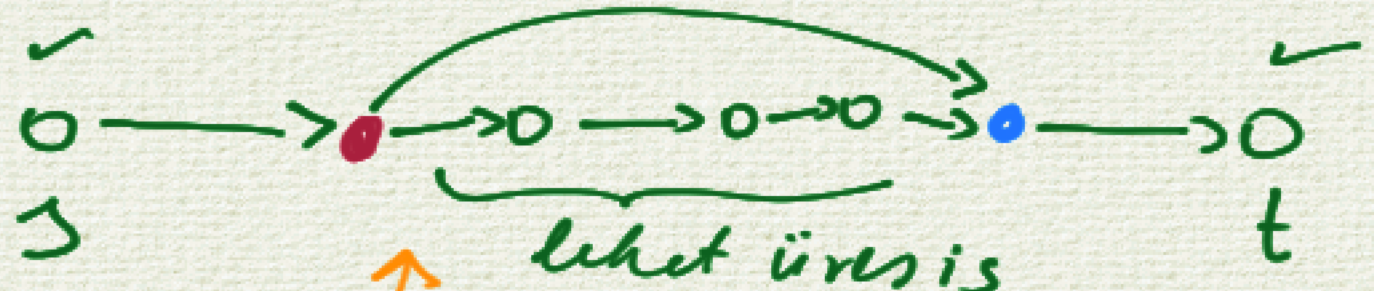
világos

ha ∀ párra a különbség ≥ 100 => ∀ pár távolsága ≥ 100 mert a nem-szomszédos párok azok még messzebb vannak a rendezett tömbben => válasz "Igen"

LSZ: $O(n \cdot \log n) + O(n)$

Egy szomszédossági mátrixával adott n csúcsú, irányított G gráfban néhány csúcs pirosra, néhány csúcs pedig kékre van színezve, a többi csúcs színtelen. A színezés egy n méretű, a csúcsokkal indexelt C tömbben adott (ha a csúcs színtelen, akkor $C[v]$ = színtelen). Adott továbbá két kijelölt színtelen csúcs, s és t és s-ből szeretnénk t-be eljutni a gráf éleit használva úgy, hogy az s után piros csúcsba megyünk, t-t kék csúcsból érjük el, az s-ből t-be vezető úton más színes csúcs nincsen és sem s-et, sem t-t nem érintjük a kiindulást és az érkezést leszámítva. Adjon $O(n^2)$ lépésszámú algoritmust, ami a szomszédossági mátrix módosításával és egy tanult algoritmus változtatás nélküli futtatásával meghatározza, hogy egy ilyen elérés során legkevesebb hány élen kell áthaladnunk (vagy ha ilyen eljutás nem lehetséges, akkor ez derüljön ki).





törlünk éleket vagy, hozzá

BFS

Ötlet:

- $O(n)$ s-ből csak pirosba lesz él, s-be nem lesz él
- $O(n)$ t-ből ne lehessen kimenni, t-be csak kékből van él
- piros csúcsba csak s-ből van él / piros csúcsból csak színtelenbe / kékbe
- kékből csak t-be / kékbe csak pirosból / színtelenből
- színtelenbe csak színtelenből / pirosból / színtelenből csak kékbe / színtelenbe

∀ csúcsra 1 sor / 1 oszlop

$2 \cdot O(n)$

$O(n)$

mátrixban

s sorában csak piros oszlopban hagyok 1-t

s oszlopában csupa 0

t sora csupa 0

piros oszlopokban csak s sorában 1-es

t oszlopában csak kék sorban 1-es

piros sorban csak sztlen / kék oszlopban 1-es

Algo

1) ez az átalakítás

$O(n^2)$ 2) BFS s-ből és ha távolság[t] = ∞ ⇒ nincs út, különben távolság[t] adja meg a keresett hosszat

jóság: 1) átalakítás után csak olyan utak maradnak, amik jók → így a BFS használható a legr. útra

LSZ: 1) $n \cdot O(n) \Rightarrow O(n^2)$ 2) $O(n^2)$