Az alábbi pszeudokód inputja egy $n \geq 2$ egész szám. A pszeudokódban lépésnek az értékadás és az összeadás számít. Igaz-e, hogy ennek a kódnak a lépésszáma $O(n^2)$? Válaszát indokolja!

```
kicsi := 1
nagy  := n
számláló := 0

ciklus amíg kicsi < nagy:
    ciklus i = 0-tól n-ig:
           számláló := számláló + 1
    ciklus vége
    kicsi : = kicsi + 1
    nagy : = nagy - 1
ciklus vége

return számláló
```

$O(1)$

$O(n)$

$\to$ n-szer fut le $O(1)$-es ciklusmag $\Rightarrow$ $O(n)$

$O(1)$

- ez a ciklus $\leq \frac{n}{2}$-szer fut le, mert $\forall$ futás után nagy − kicsi 2-vel csökken és eredetileg $n-1$ és leáll, ha kicsi $\geq$ nagy
- magja: $O(n)$-es belső ciklus $+ O(1) \Rightarrow O(n)$

$\leq \frac{n}{2} \cdot O(n)$ lépés $\Rightarrow O(n^2)$

Egész: $O(1) + O(n^2) \Rightarrow O(n^2)$ ✓

2. A $8, 1, 20, x$ tömböt összefésüléses rendezéssel rendezzük, ahol x egy olyan egész szám, ami máshol nem szerepel a tömbben. Hány összehasonlítás történhet a rendezés teljes futása alatt x értékétől függően? Az összehasonlítások darabszámának összes lehetséges értékére adjon meg egy olyan x számot, amikor ennyi összehasonlítás van (és magyarázza is el, hogy miért ennyi) és mutassa meg, hogy több vagy kevesebb összahasonlítás miért nem lehet.

$8, 1 \mid 20, x$

$8, 1$ — 1 öh $\Rightarrow$ $\boxed{1, 8}$

$20, x$ — 1 öh $\Rightarrow$ $\boxed{x, 20}$ (ha $x < 20$) VAGY $20, x$ ($x > 20$)

$\longrightarrow$ eddig 2 öh

1. eset: $x > 20$ (pl $x = 21$): $1,8$ és $20, x$ öfésülése

1 öh 20
8 öh 20

$\longrightarrow 1, 8, \underbrace{20, x}_{\text{öh nélkül}}$

2 öh $\Rightarrow$ összesen 4 öh (pl $x = 21$)

2. eset: $\boxed{x < 20}$: $1,8$ és $x,20$ öfésülése

$x < 1$:

$1,8$ és $x,20$

x öh 1
1 öh 20
8 öh 20

$\longrightarrow x, 1, 8, 20$ (20: öh nélkül)

+3 öh $\Rightarrow$ összesen 5

pl $x = -17$

$1 < x < 8$:

$1,8$ és $x,20$

1 öh x
8 öh x
8 öh 20

$\longrightarrow 1, x, 8, 20$

+3 öh $\Rightarrow$ összesen 5

$8 < x < 20$:

$1,8$ és $x,20$

1 öh x
8 öh x

$\longrightarrow 1, 8, \underbrace{x, 20}_{\text{öh nélkül}}$

+2 öh,

$\Sigma = 4$ öh

Egy 11 méretű hash táblába 7 kulcsot szúrtunk be nyílt címzéssel, lineáris próbával, majd a beszúrások után egy értéket töröltünk és így az alábbi táblát kaptuk, a törölt elemet $\star$ jelzi. A használt hash függvény a $h(K)$ = K maradéka 11-gyel osztva függvény volt. (A lineáris próba lefele indul.)

0	1	2	3	4	5	6	7	8	9	10
	6	14	$\star$	4	28	7	18			

(a) Lehetséges-e, hogy a 28 később került beszúrásra, mint a 14?
(b) Lehetséges-e, hogy a 28 később került beszúrásra, mint a 6?
Válaszait indokolja!

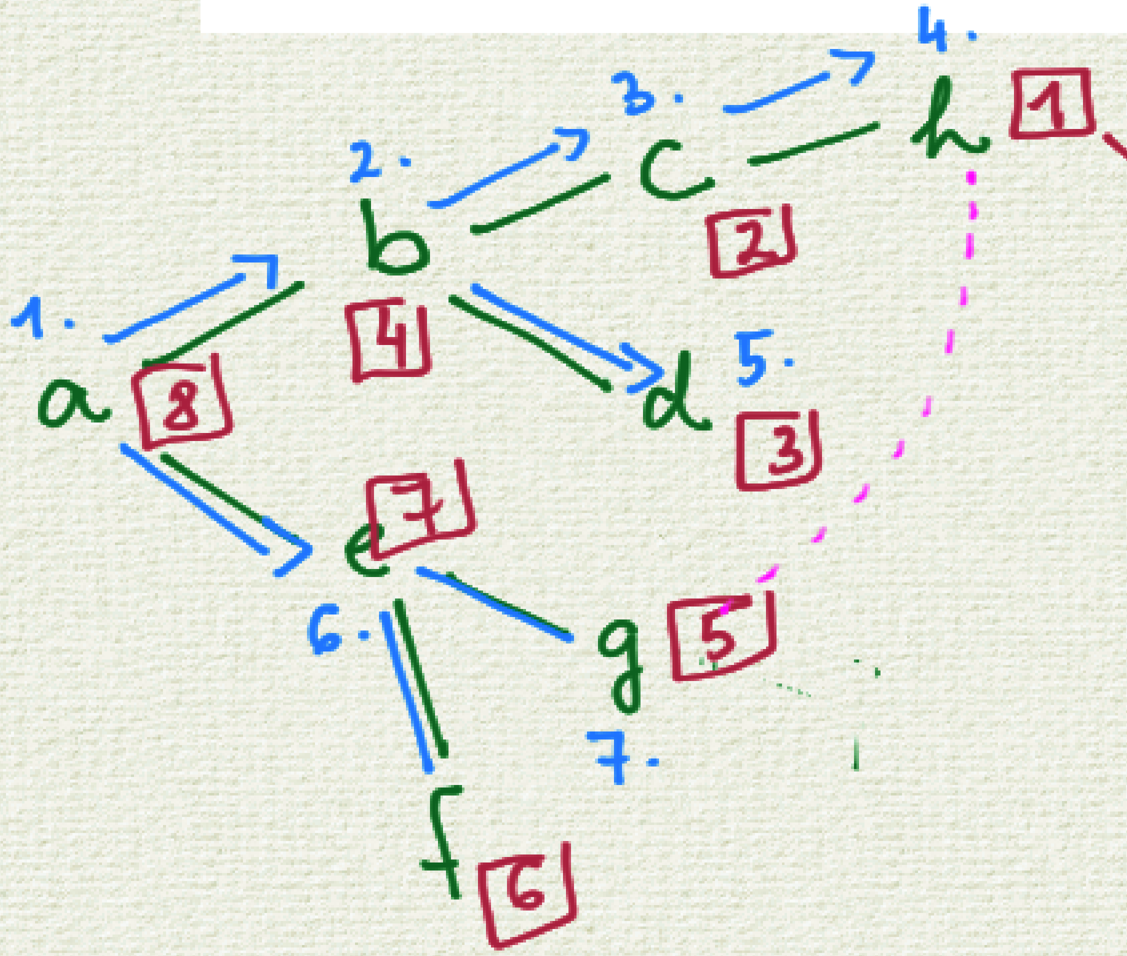
$h(K), h(K)-1, h(K)-2, \dots$

a) Igen, pl: 3 (majd töröljük), 14, 4, 18, 7, 28, 6

0	1	2	3	4	5	6	7	8	9	10
	6	14	~~3~~	4	28	7	18			

b) Nem, mert ha 6 a 28 előtt szúródik be => 5-ös cella még üres => 6 oda kerülne.

Egy nyolc csúcsú irányítatlan G gráfban a DFS (mélységi 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 bejárás során a DFS fába az alábbi élek kerülnek be ebben a sorrendben: $ab, bc, ch, bd, ae, eg, ef$.
(a) Rajzolja fel a bejáráshoz tartozó DFS fát.
(b) Adja meg a csúcsokhoz tartozó befejezési számokat.
(c) Lássa be, hogy a G gráfban a h csúcs fokszáma legfeljebb 3 lehet.
Válaszait indokolja!



b) mélységi számok bef számok

→ mert h-ból vissza kellett fordulni; hiszen következő él nem h-ból megy ide úgyse menne

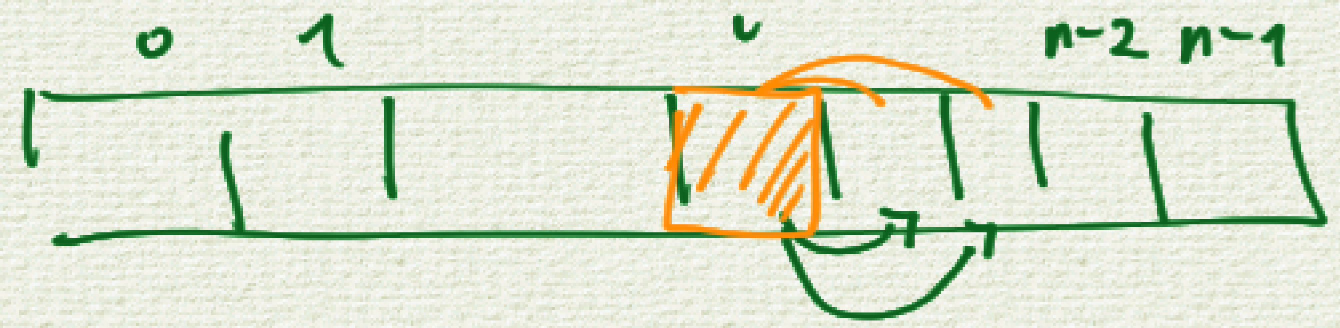
c) h csak a, b, c -vel lehet összekötve, mert ha d, e, g, f - be lenne él akkor h-ból odamentem volna és nem fordulok vissza.

1, 3, 6, 7, 8, 15

LSZ

Adott egy $n \geq 2$ méretű tömb, melyben csupa különböző egész számot tárolunk. Adjon $O(n \log n)$ lépésszámú algoritmust, ami eldönti, hogy van a tömbben három egymást követő egész szám. (Például 8, 1, 7, 3, 15, 6 jó input, mert szerepel benne 6, 7, 8.)

jóság: rendezés után az egymást követő hármasok egymás mellett vannak



Algo

1) rendezzük a tömböt öf rendezéssel $O(n \cdot \log n)$

2) Ciklus $i = 0$-tól $(n-3)$-ig:

ha $A[i+1] == A[i]+1$ és $A[i+2] == A[i]+2$:

return "Van"

c. v.

return "Nincs"

$\leq n$-szer fut le

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

Egy szomszédossági mátrixával adott n csúcsú, irányított G gráfban mindegyik csúcs vagy pirosra vagy kékre van színezve, a színezés egy n méretű, a csúcsokkal indexelt C tömbben adott. Adott továbbá két csúcs, s és t és s-ből szeretnénk t-be eljutni a gráf éleit használva úgy, hogy az úton a színek felváltva vannak. Adjon $O(n^2)$ lépésszámú eljárást, ami a szomszédossági mátrix módosításával és egy tanult algoritmus változtatás nélküli futtatásával eldönti, hogy van-e ilyen út vagy nincsen.

BFS | DFS

LSZ

jóság

Algo: 1) módosítjuk a gráfot: csak piros → kék és kék → piros élek legyenek:

kivesszük $\forall$ p → p, k → k élt ←

$\underbrace{\text{végig a mátrixon}}_{n^2 \text{ cella}}$ és ha $A[i,j] == 1$ és $C[i] == C[j]$ ⇒ $A[i,j] := 0$

$\underbrace{n^2 \text{ cella} \longrightarrow \forall \text{ cella } O(1)}_{O(n^2)}$ →

⇓

megmaradó utakon felváltva jönnek a színek ⇒ új G'-ben elérhető ⟺ G-ben felváltott úton elérhetőség

2) BFS | DFS s-ből az új gráfban: bejárva[t] $\overset{?}{==}$ 1

igen → t elérhető felváltott színű úton

nem → nem érhető el

$O(n^2)$

mert BFS | DFS is → mutatja, hogy vmi elérhető-e