Algoritmusok és gráfok - Mintavizsga
2019. ősz

1. Az alábbi pszeudokód inputja két, egész számokat tartalmazó n méretű tömb, A és B. Mutassa meg, hogy a pszeudokód által leírt algoritmus lépésszáma $O(n^2)$.

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

2. A Dijkstra algoritmus lényegi része egy amíg-ciklus, mely két részlet (a két üresen hagyott téglalap) kivételével így néz ki:

```
ciklus amíg van KÉSZ-en kívüli csúcs, ahol d[v] nem végtelen:
    v* := az a csúcs, ahol d[v] minimális
    v* KÉSZ-be megy
    távolság[v*] := d[v*]
    d[v*] : = *
    ciklus w = 1-tól n-ig:
        ha A[v*,w] nem végtelen és d[w] nem *:
            ha távolság[v*] + c(v*, w) < d[w]:
                d[w] := [          ]
                honnan[w] := [     ]
    ciklus vége
ciklus vége
```

   Egészítse ki a kódot a két üres téglalap kitöltésével és magyarázza el röviden (2-3 mondatban), hogy miért így kell a két hiányzó értéket meghatározni.

3. A G irányított, élsúlyozott gráf élei a következők (zárójelben az élsúlyok):
   $ab(3), ac(-5), ae(1), bc(1), bd(4), be(-2), ce(7), de(-4)$.
   Ebben a gráfban az a, b, c, d, e sorrend egy topologikus sorrend, ezt használva a tanult algoritmussal már meghatároztuk az a-ból az a, b, c, d csúcsokba vezető legrövidebb utak hosszát: *távolság[a] = 0*, *távolság[b] = 3*, *távolság[c] = -5*, *távolság[d] = 7*. Hogyan kapjuk meg *távolság[e]*-t a tanult eljárással ezekből az adatokból?

4. Mutasson példát olyan 5 csúcsú összefüggő, irányítatlan, élsúlyozott gráfra és benne olyan kezdőcsúcsra, ahol a gráfban az élsúlyok mind különbözőek és ahol a minimális feszítőfa keresésére tanult Prim algoritmus végére nem a 4 legkisebb súlyú él kerül be minimális feszítőfába.

5. Indokolja meg, hogy miért nem lehetséges, hogy egy bináris keresőfában egy keresés során az alábbi számokat járjuk be ebben a sorrendben: $13, 10, 8, 9, 12$.

6. A G irányított gráf élei a következők: $ab, ac, bd, be, cd, cg, df, ed, gd$.
   Járja be ezt a gráfot mélységi bejárással (DFS) az a csúcsból kiindulva úgy, hogy ha a futás során választási lehetőség adódik, akkor mindig az ábécé szerint korábban levő csúcsba megyünk.
   A kapott befejezési számok segítségével az órán tanult módszerrel állapítsa meg, hogy ez a gráf DAG-e vagy sem.

7. Egy szomszédossági mátrixával adott n csúcsú irányítatlan G gráfban adott a csúcsoknak egy $u_1, u_2, \ldots, u_n$ sorrendje (tudjuk, hogy itt minden csúcs pontosan egyszer szerepel, ezt nem kell ellenőrizni). Azt szeretnénk eldönteni, hogy a csúcsok ebben a sorrendben kört alkotnak-e a gráfban. Adjon erre a feladatra $O(n)$ lépésszámú algoritmust.

8. Szomszédossági mátrixával adott egy n csúcsú irányítatlan gráf, mely emberek (kölcsönös) ismeretségeit ábrázolja: a csúcsok az emberek és akkor van él két csúcs között, ha az emberek ismerik egymást. Adott két ember, A és B, akik nem ismerik egymást. Azt szeretnénk eldönteni $O(n^2)$ lépésben, hogy vannak-e olyan E_1 és E_2 emberek, hogy A ismeri E_1-t, E_1 ismeri E_2-t, E_2 pedig ismeri B-t. (Az is jó, ha E_1 és E_2 ugyanaz az ember, azaz ekkor A-nak és B-nek van közös ismerőse.)
   Melyik tanult algoritmust lehet alkalmazni erre a feladatra és hogyan, hogy megválaszoljuk ezt a kérdést?

Algoritmusok és gráfok - Vizsga
2020. január 8.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

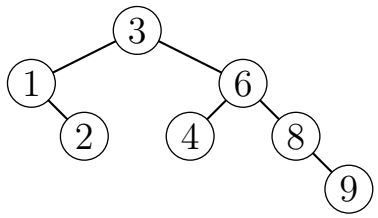
1. Lássa be megfelelő c konstans és n_0 küszöbérték megadásával, hogy ha egy algoritmus lépésszáma

$$2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n} - 2019$$

akkor az algoritmus futásideje $O(n^3)$.

2. (a) Írja le, hogy irányítatlan gráfban mit jelent az, hogy egy v csúcs $d(v)$-vel jelölt fokszáma 8.
(b) Mondja ki az irányítatlan gráfokban a fokszámok összegéről tanult tételt és bizonyítsa be.

3. (a) Az órán tanult eljárásokkal szúrja be az alábbi bináris keresőfába a 7-et, majd törölje ki az így kapott fából a 3-at. Mindkét műveletnél magyarázza is el 1-2 mondatban, hogy mi miért történt.
(b) AVL-fa-e a kiindulási fa és AVL-fa-e a két művelet után kapott fa?



4. Egy hat csúcsú gráf csúcsai meg vannak címkézve a $2, 3, 4, 5, 6, 7$ számokkal és a gráfban akkor van él két csúcs között, ha a csúcsok címkéi relatív prímek (azaz a legnagyobb közös osztójuk 1). A gráf élei súlyozottak, az i és j címkéjű csúcsok közt futó él súlya $i \cdot j$ (azaz például a 6 és 7 címkéjű csúcsok között van él és ennek súlya 42).
Minimális súlyú feszítőfát keresünk ebben a gráfban Kruskal algoritmusával. Melyik éleket választja be az algoritmus és milyen sorrendben?

5. Dijkstra algoritmusát használjuk az A, B, C, D, E csúcsokból álló irányított, élsúlyozott G gráfban az A kezdőcsúcsból, eközben a d tömb így változik (az egyes sorok a d tömb változását mutatják egy-egy csúcs KÉSZ halmazba kerülése után).

A	B	C	D	E
*	5	∞	1	∞
*	3	5	*	∞
*	*	4	*	∞
*	*	*	*	8
*	*	*	*	*

Milyen csúcsokkal van biztosan összekötve a B csúcs (bejövő és kimenő élekre is gondoljon) és mi ezeknek az élsúlya?

6. DFS-t (mélységi bejárást) futtatunk az A, B, C, D, E, F csúcsokból álló irányítatlan G gráfon a D kezdőcsúcsból, a DFS fába a DA, AF, FE, DB, BC élek kerülnek be ebben sorrendben. Mely csúcsokkal lehet összekötve az F csúcs a gráfban és melyekkel nem?

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf és adott egy, a csúcsokkal indexelt R tömbben a csúcsok egy részhalmaza oly módon, hogy ha a v csúcs nincsen benne a részhalmazban, akkor $R[v] = 0$, ha pedig v benne van a részhalmazban, akkor $R[v] = 1$.
   Azt szeretnénk eldönteni, hogy igaz-e, hogy a gráf minden élének legalább egyik végpontja az R halmazban van.

   Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

8. Szomszédossági mátrixával adott egy n csúcsú, irányított, élsúlyozott G gráf csupa pozitív élsúllyal, ahol minden csúcsnak van egy címkéje, egy pozitív egész szám, ezek a címkék nem feltétlenül különbözőek. A csúcsok címkéi egy, a csúcsokkal indexelt T tömbben adottak. Ki van jelölve még két csúcs is, A és B, a gráfban. Adjon $O(n^2)$ lépésszámú algoritmust, ami megtalálja a legrövidebb összhosszúságú olyan utat A-ból B-be ahol az úton csak 100-nál kisebb címkéjű csúcsokon haladunk át (A és B címkéjét is beleértve) vagy jelzi, ha nincsen ilyen út.

Algoritmusok és gráfok - Vizsga
2020. január 15.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

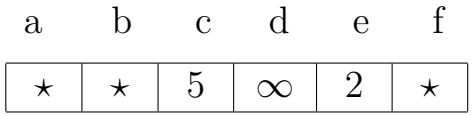
1. Az alábbi pszeudokód inputja egy n sorból és n oszlopból álló A mátrix, lépésnek az értékadás és az összehasonlítás számít. Igaz-e, hogy ez a kód $O(n)$-es? Ha igaz, akkor lássa be ezt, ha pedig nem igaz, akkor lássa be, hogy $O(n^2)$-es az algoritmus.

```
ciklus i = 1-től n-ig:
    ha A[i,i] == 0:
        k := 0
        ciklus amíg k < 20:
            A[i,i] := A[i,i] + k
            k := k+1
        ciklus vége
ciklus vége
```

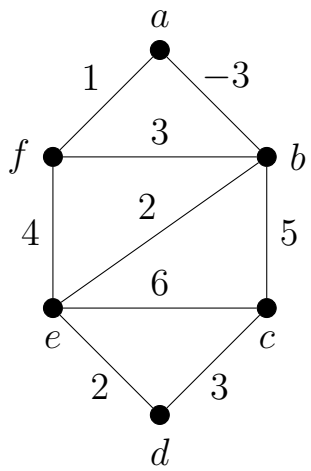
2. (a) Mikor mondjuk, hogy egy irányítatlan gráf fa?
   (b) Mondja ki az n csúcsú fa éleinek számáról tanult állítást és bizonyítsa be. A bizonyítás során felhasználhatja indoklás nélkül, hogy minden legalább 2 csúcsú fában van legalább két elsőfokú pont.

3. Egy irányított gráf élei a következők: $AB, AC, BC, BD, CE, DF, DG, EB, ED, EF$. Futassa le a mélységi bejárást az A csúcsból ezen a gráfon, majd a mélységi bejárás eredményét felhasználva, az órán tanult módszerrel döntse el, hogy van-e ebben a gráfban topologikus sorrend és ha van, akkor adjon is meg egyet.

4. Az alábbi gráfon futtatjuk Prim algorimusát az a csúcsból, az ab és af élek beválasztása után a *legolcsóbb* tömb így néz ki:



a	b	c	d	e	f
⋆	⋆	5	∞	2	⋆



   (a) Hogyan néz ki a *közeli* tömb ekkor és miért?
   (b) Melyik élet választja be most az algoritmus és hogyan változik ettől a *legolcsóbb* tömb? Röviden indokolja is a választ.

5. BFS-t (szélességi bejárást) futtatunk az A, B, C, D, E, F csúcsokból álló irányítatlan G gráfon az A kezdőcsúcsból, a BFS fába az AB, AC, BD, CE, CF élek kerülnek be ebben sorrendben. Mely csúcsokkal lehet összekötve az E csúcs a gráfban, melyekkel nem és miért?

6. Mutassa meg, hogy ha egy 30 csúcsú AVL-fában az 1 és 30 közötti egész számokat tároljuk (az 1 és a 30 is benne van), akkor nem lehetséges, hogy egy érték keresése során a $2, 8, 3, 7$ számokat látjuk ebben a sorrendben.

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf és adott egy, a csúcsokkal indexelt R tömbben a csúcsok egy részhalmaza oly módon, hogy ha a v csúcs nincsen benne a részhalmazban, akkor $R[v] = 0$, ha pedig v benne van a részhalmazban, akkor $R[v] = 1$.
   Azt szeretnénk eldönteni, hogy igaz-e, hogy mindegyik R-beli csúcspár össze van kötve a gráfban.

   Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

8. Szomszédossági mátrixával adott egy város úthálózatának élsúlyozott, irányított gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak, az élek súlya pedig azt mutatja, hogy mennyi az adott útszakasz hossza. Adott a gráfban két kijelölt él, (u_1, v_1) és (u_2, v_2), és két kijelölt csúcs, A és B, és szeretnénk meghatározni a legrövidebb olyan út hosszát, ami A-ból B-be vezet úgy, hogy legfeljebb az egyik kijelölt élet használja.

   Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk oldani ezt a feladatot? (Itt n szokásos módon a gráf csúcsainak számát jelöli.)

Algoritmusok és gráfok - Vizsga
2020. január 22.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

1. Az alábbi függvények közül pontosan egyre igaz, hogy $O(n^2)$-es. Válassza ki ezt a függvényt és lássa be megfelelő c konstans és n_0 küszöbérték megadásával, hogy $O(n^2)$-es.

$$\log(n^2) + \tfrac{n(n-1)(n-2)}{2010} \qquad 2020n \log n + \frac{2^n \cdot n(n-1)}{3^n} \qquad 20n^2(\log n)^2 - 8$$

2. A Prim algoritmus lényegi része egy amíg-ciklus, mely két részlet (a két üresen hagyott téglalap) kivételével így néz ki:

```
ciklus amíg van olyan csúcs, ahol legolcsóbb[v] nem *:
    v* := az a csúcs, ahol legolcsóbb[v] minimális
    v* LEFEDVE-be megy
    legolcsóbb[v*] := *
    (közeli[v*], v*) F-be kerül
    ciklus w = 1-tól n-ig:
        ha A[v*,w] nem végtelen és legolcsóbb[w] nem *:
            ha c(v*, w) < legolcsóbb[w]:
                legolcsóbb[w] := [          ]
                közeli[w] := [          ]
    ciklus vége
ciklus vége
```

(a) Mit tárol `legolcsóbb[w]` egy w csúcs esetén?
(b) Mit tárol `közeli[w]` egy w csúcs esetén?
(c) Egészítse ki a kódot a két üres téglalap kitöltésével és magyarázza el röviden (2-3 mondatban), hogy miért így kell a két hiányzó értéket meghatározni.

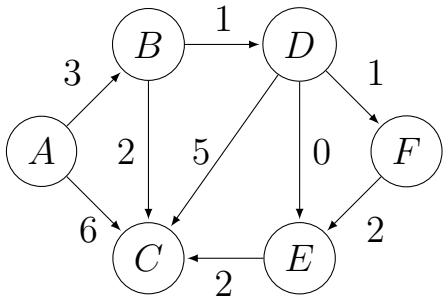
3. Egy kezdetben üres, 11 méretű hash táblába nyílt címzéssel, lineáris próbával szúrtunk be hét egész számot, a használt hash függvény a $h(K) = \text{K}$ maradéka 11-gyel osztva függvény volt. A beszúrások után egy számot kitöröltünk a táblából, a törölt elem helyét * jelzi, az eljárás végén az alábbi táblát kaptuk:

0	1	2	3	4	5	6	7	8	9	10
		14	3		5	17	*	8	20	

(a) Szúrja be ebbe a táblába a 31-et, majd törölje ki a 17-et. Adja meg a tábla állapotát mindkét művelet után és mindkét esetben jelezze, hogy melyik cellákat érintettük.
(b) Mely cellákat érintjük a 19 keresése során az (a) rész végén kapott táblában?

4. Az alábbi gráfon futtatjuk Dijkstra algorimusát az A csúcsból, az A és B csúcsok KÉSZ-be kerülése után a d tömb így néz ki:

A	B	C	D	E	F
$\star$	$\star$	5	4	∞	∞



(a) Magyarázza el röviden, hogy $d(F)$ miért végtelen és $d(D)$ miért 4.
(b) Melyik csúcs kerül a következő lépésben a KÉSZ halmazba és miért?
(c) Hogyan néz ki a d tömb az új csúcs KÉSZ-be kerülése miatti módosítások után és miért?

5. Az $5, 2, x, 10$ tömb összefésüléses rendezése során 4 összehasonlítás történik. Adja meg x összes lehetséges értékét, ha tudjuk, hogy x olyan egész szám, ami máshol nem szerepel a tömbben. Válaszát indokolja is.

6. Egy G irányított, élsúlyozott gráfban az a, b, c, d, e sorrend topologikus sorrend. Ezen topologikus sorrendet használva a tanult algoritmust futtatjuk az a-tól vett legrövidebb utak hosszának meghatározására. Az a, b, c, d csúcsokra ezeket a távolságokat kaptuk: *távolság*[a] = 0, *távolság[b]* = 2, *távolság[c]* = -7, *távolság[d]* = 5.
(a) Miért biztos, hogy van a gráfban ab él? Mi ennek a súlya és miért?
(b) Ha van a gráfban cd él, akkor mi lehet ennek a súlya és miért?

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf. Adjon $O(n^2)$ lépésszámú algoritmust, ami eldönti, hogy van-e a gráfban olyan másodfokú csúcs, aminek a két szomszédja között van él a gráfban.

8. Szomszédossági mátrixával adott egy n csúcsú, irányított G gráf és benne egy kijelölt (u, v) irányított él. Szeretnénk megtalálni a legkevesebb élből álló, az (u, v) élen átmenő irányított kört a gráfban.

   Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk oldani ezt a feladatot?

Algoritmusok és gráfok - Vizsga
2020. január 29.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

1. Az alábbi pszeudokód inputja egy $n \geq 2$ hosszú A tömb. Igaz-e, hogy ez a kód $O(n)$-es, ha egy lépésnek az összehasonlítás, az összeadás és az értékadás számít? Ha igaz, akkor lássa be ezt, ha pedig nem igaz, akkor lássa be, hogy $O(n^2)$-es.

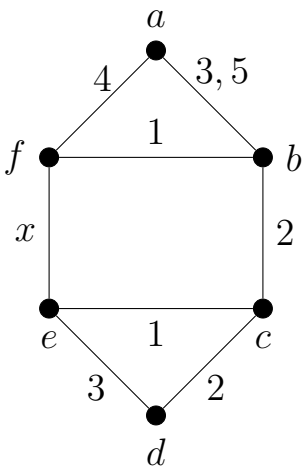
```
ciklus i = 0-tól (n-1)-ig:
    ha A[i] > 0:
        ciklus amíg A[i] < 5:
            A[i] := A[i] + 1
        ciklus vége
ciklus vége
```

2. (a) Adja meg a bináris keresőfa definícióját (elég a bináris keresőfa tulajdonságot leírnia, nem kell definiálni azt, hogy mi a bináris fa).
(b) Írja le, hogy hogyan kell beszúrni egy új elemet bináris keresőfába. Mennyi ennek az eljárásnak a lépésszáma és miért?

3. Egy irányított gráf élei a következők: $AB, AC, BG, CB, CD, CG, DA, DE, DF, EF, GD, GF$. Ebben a gráfban BFS-t (szélességi bejárást) futtatunk az A kezdőcsúcsból úgy, hogy ha bármikor választási lehetőség adódik, akkor az ábécé szerint előbb levő csúcsot választjuk.
(a) Milyen sorrendben járjuk be a csúcsokat és mi lesz a kapott BFS fa? (Indokolni ezt nem kell, elég a sorrend és a fa megadása.)
(b) Hogyan néz ki a *honnan* tömb a BFS futásának a végén? Indoklásképpen írja le röviden, hogy mit tárol a *honnan* tömb.
(c) Adja meg a Q sor állapotát a kód futásának kezdetén, majd minden egyes csúcs bejárása után. Indoklásképpen röviden írja le, hogy a Q sornak mi a szerepe a kódban.

4. A G irányított, élsúlyozott gráf élei a következők (zárójelben az élsúlyok):
$ac(8), af(7), bf(11), cd(10), ce(7), cf(6), eb(-5), ed(2), ef(3)$.
Ebben a gráfban az órán tanult, DAG-ban leghosszabb utat kereső algoritmust futtatjuk a-ból az a, c, e, b, d, f topologikus sorrendet használva. Az a, c, e, b, d csúcsokba vezető leghosszabb utak hosszát már meghatároztuk: *leghosszabb[a] = 0*, *leghosszabb[c] = 8*, *leghosszabb[e] = 15*, *leghosszabb[b] = 10*, *leghosszabb[d] = 18*. Hogyan kapjuk meg *leghosszabb[f]*-et a tanult eljárással ezekből az adatokból? A megoldásából látszódjon, hogy hogyan használja a tanult eljárást.

5. Az alábbi gráfon, ahol az x élsúly nem ismert, futtatva Kruskal algorimusát az első három él, amit az eljárás beválaszt: bf, ec, bc. Adja meg x összes lehetséges értékét (röviden indokolva válaszát) és írja le, hogy mely éleket és milyen sorrendben választja be ezután az algoritmus és miért.



6. Egy kezdetben üres, 11 méretű hash táblába nyílt címzéssel, lineáris próbával szúrtunk be nyolc egész számot, a használt hash függvény a $h(K) = \text{K}$ maradéka 11-gyel osztva függvény volt. A beszúrások után két számot kitöröltünk a táblából, a törlések helyét $*$ jelzi. Mutassa meg, hogy miért nem lehetséges, hogy az eljárás végén az alábbi táblát kaptuk:

0	1	2	3	4	5	6	7	8	9	10
16	*		14	6	*	17			9	21

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf és adott egy, a csúcsokkal indexelt R tömbben a csúcsok egy részhalmaza oly módon, hogy ha a v csúcs nincsen benne a részhalmazban, akkor $R[v] = 0$, ha pedig v benne van a részhalmazban, akkor $R[v] = 1$.
   Azt szeretnénk eldönteni, hogy igaz-e, hogy egyetlen él sem fut R-beli csúcsok közt a gráfban.

   Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

8. Egy A szomszédossági mátrix-szal adott egy város úthálózatának élsúlyozott, irányított gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak, az élek súlya pedig azt mutatja, hogy mennyi az átlagos idő, ami az út megtételéhez autóval szükséges. Adott továbbá egy másik B mátrix is, ami azt adja meg minden közvetlen útra, hogy mekkora a legnagyobb magasságú autó, amivel elakadás nélkül át lehet haladni az adott szakaszon: a B mátrix i-edik sorának, j-edik oszlopában álló $B[i, j]$ érték mutatja a legnagyobb magasságot, amivel még megtehető az i csomópontból a j csomópontba vezető közvetlen út.

   Adott a gráfban két kijelölt csúcs, X és Y. Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk találni a leggyorsabban megtehető olyan utat X-ből Y-ba, amit legfeljebb 350 centiméter magas autóval meg lehet tenni? (Itt n szokásos módon a gráf csúcsainak számát jelöli.)

Algoritmusok és gráfok - Vizsga
2020. január 8.
Megoldások

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

1. Lássa be megfelelő c konstans és n_0 küszöbérték megadásával, hogy ha egy algoritmus lépésszáma

$$2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n} - 2019$$

akkor az algoritmus futásideje $O(n^3)$.

Megoldások

Olyan c pozitív konstanst és egy olyan n_0 küszöbindexet kell találnunk, melyre igaz, hogy

$$2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n} - 2019 \leq c \cdot n^3, \text{ha } n \geq n_0$$

Ilyen c és n_0 értéket például az alábbi egyenlőtlenség-sorozattal tudunk találni:
$2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n} - 2019 \leq 2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n}$
mert a negatív tagot el lehet hagyni, tetszőleges $n \geq 1$ esetén, továbbá
$2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n} \leq 2020n^3 + \frac{(n-1)(n-2)(n-3)(n-4)}{n}$
mert $\log n \leq n$ tetszőleges $n \geq 1$ esetén, továbbá
$2020n^3 + \frac{(n-1)(n-2)(n-3)(n-4)}{n} \leq 2020n^3 + (n-2)(n-3)(n-4)$
mert $\frac{(n-1)}{n} \leq 1$ tetszőleges $n \geq 1$ esetén és
$2020n^3 + (n-2)(n-3)(n-4) \leq 2020n^3 + n^3$
mert $(n-2) \leq n, (n-3) \leq n, (n-4) \leq n$.

Azt kaptuk tehát, hogy
$2020n^2 \log n + \frac{(n-1)(n-2)(n-3)(n-4)}{n} - 2019 \leq 2021n^3$ tetszőleges $n \geq 1$ esetén, vagyis $c = 2021$ és $n_0 = 1$ jó választás.

2. (a) Írja le, hogy irányítatlan gráfban mit jelent az, hogy egy v csúcs $d(v)$-vel jelölt fokszáma 8.
(b) Mondja ki az irányítatlan gráfokban a fokszámok összegéről tanult tételt és bizonyítsa be.

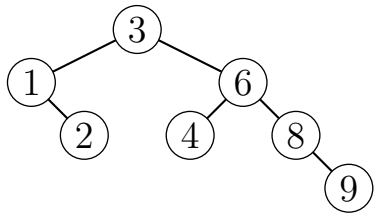
Megoldás

(a) Egy csúcs fokszáma irányítatlan gráfban a csúcsra illeszkedő élek száma, azaz $d(v) = 8$ azt jelenti, hogy a v csúcsra 8 él illeszkedik.
(b) Irányítatlan gráfban a fokszámok összege $2e$, azaz

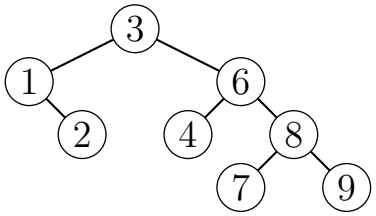
$$\sum_{v \in V} d(v) = 2e$$

mert amikor a fokszámokat összeadjuk, akkor minden élet kétszer számolunk, az él két végpontjánál.

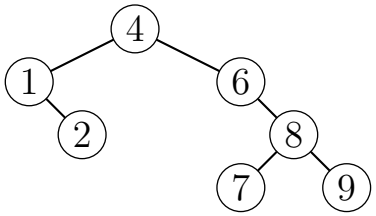
3. (a) Az órán tanult eljárásokkal szúrja be az alábbi bináris keresőfába a 7-et, majd törölje ki az így kapott fából a 3-at. Mindkét műveletnél magyarázza is el 1-2 mondatban, hogy mi miért történt.
(b) AVL-fa-e a kiindulási fa és AVL-fa-e a két művelet után kapott fa?



Megoldás
(a)



Mert a 7 nagyobb a 3-nál és 6-nál (jobbra megyünk), de kisebb a 8-nál (balra megyünk).



Mert a jobb részfa legkisebb elemét, azaz a 4-et kell a 3 helyére rakni.

(b) A kezdeti fa AVL-fa, mert minden csúcsra igaz, hogy a bal és jobb részfájának magassága maximum 1-gyel tér el, de a végén kapott fa nem AVL-fa, mert a 6 bal részfája üres, azaz 0 magasságú, a jobb fája meg kettő magas.

4. Egy hat csúcsú gráf csúcsai meg vannak címkézve a $2, 3, 4, 5, 6, 7$ számokkal és a gráfban akkor van él két csúcs között, ha a csúcsok címkéi relatív prímek (azaz a legnagyobb közös osztójuk 1). A gráf élei súlyozottak, az i és j címkéjű csúcsok közt futó él súlya $i \cdot j$ (azaz például a 6 és 7 címkéjű csúcsok között van él és ennek súlya 42).
Minimális súlyú feszítőfát keresünk ebben a gráfban Kruskal algoritmusával. Melyik éleket választja be az algoritmus és milyen sorrendben?

Megoldás
Kruskal algoritmusa élsúly szerint növekvő sorban nézi az éleket és pontosan akkor

választ be egy élet, ha az nem alkot kört a már beválasztott élekkel.
A gráf élei élsúly szerint növekvően: c(2,3) = 6, c(2,5) = 10, c(3,4) = 12, c(2,7) = 14, c(3,5) = 15, c(4,5) = 20, c(3,7) = 21, c(4,7) = 28, c(5,6) = 30, c(5,7) = 35, c(6,7) = 42.
Végigmegyünk ezen a listán és először bekerül a 6-os, a 10-es , 12-es és 14-es él, mert ezek egyike sem alkot kört a korábbiakkal. A 15-ös, 20-as, 21-es és 28-as él kört csinálna, ezeket nem vesszük be, de 30-t igen és ezzel kész is a fa.

5. Dijkstra algoritmusát használjuk az A, B, C, D, E csúcsokból álló irányított, élsúlyozott G gráfban az A kezdőcsúcsból, eközben a d tömb így változik (az egyes sorok a d tömb változását mutatják egy-egy csúcs KÉSZ halmazba kerülése után).

A	B	C	D	E
*	5	∞	1	∞
*	3	5	*	∞
*	*	4	*	∞
*	*	*	*	8
*	*	*	*	*

Milyen csúcsokkal van biztosan összekötve a B csúcs (bejövő és kimenő élekre is gondoljon) és mi ezeknek az élsúlya?

Megoldás
Az első sorból látszik, hogy A-ból van egy él a B-be 5 súllyal, mert az első sor a kezdőcsúcsból kimenő élek súlyait tartalmazza.
Az első és a második sor közti változásból látható, hogy ekkor D került be a KÉSZ-be 1 távolsággal és emiatt lett B értéke 3, azaz biztosan van él D-ből B-be, 2 súllyal.
A második és harmadik sor közti változásból látható, hogy ekkor B került be a KÉSZ-be 3 távolsággal és emiatt lett C értéke 4, azaz a 3 távolságú B-n át van egy rövidebb út C-be, 1 súllyal.

6. DFS-t (mélységi bejárást) futtatunk az A, B, C, D, E, F csúcsokból álló irányítatlan G gráfon a D kezdőcsúcsból, a DFS fába a DA, AF, FE, DB, BC élek kerülnek be ebben sorrendben. Mely csúcsokkal lehet összekötve az F csúcs a gráfban és melyekkel nem?

Megoldás
A-val biztosan össze van kötve F, hiszen van a DFS fában AF él.
B-vel nem lehet összekötve F, mert ha lenne BF él, akkor az FE él bejárása után nem lépnénk vissza F-ből A-ba, hanem az FB éllel felfedeznénk B-t.
A C csúcs ugyanezért nem lehet összekötve F-fel: ha lenne CF él, akkor az FE él bejárása után nem lépnénk vissza F-ből A-ba, hanem az FC éllel felfedeznénk C-t.
D és F között lehet él, akkor is futhat ugyanígy a mélységi bejárás (azaz ekkor is a DA, AF, FE, DB, BC élek választódnak be ebben a sorrendben), mert F éleinek vizsgálatakor D már be van járva és így mindegy, hogy vezet-e oda él vagy sem.
E-vel biztosan össze van kötve F, hiszen van a DFS fában FE él.

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf és adott egy, a csúcsokkal indexelt R tömbben a csúcsok egy részhalmaza oly módon, hogy ha a v csúcs nincsen benne a részhalmazban, akkor $R[v] = 0$, ha pedig v benne van a részhalmazban, akkor $R[v] = 1$.
Azt szeretnénk eldönteni, hogy igaz-e, hogy a gráf minden élének legalább egyik végpontja az R halmazban van.

   Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

   Megoldás
   Végigmegyünk a szomszédossági mátrix elemein (két egymásba ágyazott ciklussal, a külső ciklus a sorokon fut, a belső pedig a sorokon belül az oszlopok indexein) és ha az i-edik sor és j-edik oszlop találkozásánál 1-t látunk (azaz van él az i-edik és j-edik csúcs között), akkor megnézzük, hogy $R[i]$ és $R[j]$ közül legalább az egyik 1-e.
   Ha végig tudunk menni a mátrixon úgy, hogy ez mindig teljesül, akkor a gráf minden élének legalább egyik végpontja az R halmazban van, egyébként (ha találunk olyan 1-es bejegyzést a máztixban valami i és j csúcs között, ahol $R[i]$ és $R[j]$ is 0), akkor nem igaz, hogy a gráf minden élének legalább egyik végpontja az R halmazban van.

   Pszeudokóddal:

```
Igaz_e := True
ciklus i =1-tól n-ig:
    ciklus j = 1-tól n-ig:
        ha A[i,j] == 1:
            ha (R[i] == 0 és R[j] ==0):
                Igaz_e : = False
    ciklus vége
ciklus vége
return Igaz_e
```

   Ez azért jó, mert $A[i, j] == 1$ ellenőrzése pont azt jelenti, hogy éleket nézem meg és pont azt ellenőrzöm, ami a kérdés, hogy vagy i vagy j benne van-e R-ben.
   A lépésszám pedig $O(n^2)$, mert a külső ciklus n-szer fut le, ennek a magja egy n-szer lefutó belső ciklus, aminek a magja $O(1)$ lépés, vagyis a belső ciklus $O(n)$, az egész eljárás pedig $O(n^2)$.

8. Szomszédossági mátrixával adott egy n csúcsú, irányított, élsúlyozott G gráf csupa pozitív élsúllyal, ahol minden csúcsnak van egy címkéje, egy pozitív egész szám, ezek a címkék nem feltétlenül különbözőek. A csúcsok címkéi egy, a csúcsokkal indexelt T tömbben adottak. Ki van jelölve még két csúcs is, A és B, a gráfban. Adjon $O(n^2)$ lépésszámú algoritmust, ami megtalálja a legrövidebb összhosszúságú olyan

utat A-ból B-be ahol az úton csak 100-nál kisebb címkéjű csúcsokon haladunk át (A és B címkéjét is beleértve) vagy jelzi, ha nincsen ilyen út.

Megoldás

Algoritmus:

1. Módosítsuk a gráfot úgy, hogy a 100 vagy annál nagyobb címkéjű csúcsokba belemenő és onnan kijövő éleket kitöröljük.
2. Ebben az új gráfban futtassuk a Dijkstra algoritmust az A csúcsból. Ha B távolsága a végén végtelen, akkor nincs jó út, egyébként pedig B távolsága a keresett érték.

A gráf módosítása pontosabban: végigmegyek a T tömbön és ha az aktuálisan vizsgált $T[v]$ érték legalább 100, akkor a gráf szomszédossági mátrixában a v-edik sor és oszlop minden értékét végtelenre állítom.

```
ciklus v =1-tól n-ig:
    ha T[v] >= 100:
        ciklus j = 1-tól n-ig:
            A[v,j] : = végtelen
            A[j,v]: = végtelen
        ciklus vége
ciklus vége
```

Lépésszám:

1. A T tömb minden értékére egyszer megyek végig egy során és egy oszlopán a mátrixnak, az $2n$ módosítás, amit n-szer kell legfeljebb megtennem, ez $n \cdot O(n) = O(n^2)$. (Vagy a pszeudokódra hivatkozva: a külső ciklus n-szer fut le, ennek a magja 1 lépés és egy belső ciklus, ami O(n)-es, mert n-szer fut le és magja $O(1)$, vagyis $n(1 + O(n)) = n \cdot O(n) = O(n^2)$).
2. A kapott mátrix is egy n csúcsú gráfhoz tartozik, ebben Dijkstra $O(n^2)$ lépésben fut, így a módosítás és a futtatás együtt $O(n^2) + O(n^2) = O(n^2)$.

Jóság: A módosítással éppen azok a csúcsok válnak elérhetetlenné, amiket nem szabad használnunk, azaz az új gráf útjai megegyeznek az eredeti gráf használható útjaival. Dijkstra algoritmusa hasznáható, mert minden élsúly pozitív.

Algoritmusok és gráfok - Vizsga
2020. január 15.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

1. Az alábbi pszeudokód inputja egy n sorból és n oszlopból álló A mátrix, lépésnek az értékadás és az összehasonlítás számít. Igaz-e, hogy ez a kód $O(n)$-es? Ha igaz, akkor lássa be ezt, ha pedig nem igaz, akkor lássa be, hogy $O(n^2)$-es az algoritmus.

```
ciklus i = 1-tól n-ig:
    ha A[i,i] == 0:
        k := 0
        ciklus amíg k < 20:
            A[i,i] := A[i,i] + k
            k := k+1
        ciklus vége
ciklus vége
```

Megoldás
Ez a kód $O(n)$-es. A külső ciklus n-szer fut le, ennek a magja pedig $O(1)$-es a következők miatt: Két lépés után a egy belső ciklus következik, ami 20-szor fut le, mert minden lefutáskor nő eggyel a k értéke. A belső ciklus magja konstans sok lépés, így a belső ciklus $O(1)$ lépésből áll, azaz a külső ciklus magja $2 + O(1)$, azaz $O(1)$, így az egész kód $O(n)$.

2. (a) Mikor mondjuk, hogy egy irányítatlan gráf fa?
(b) Mondja ki az n csúcsú fa éleinek számáról tanult állítást és bizonyítsa be. A bizonyítás során felhasználhatja indoklás nélkül, hogy minden legalább 2 csúcsú fában van legalább két elsőfokú pont.

Megoldás
(a) Egy irányítatlan gráf akkor fa, ha összefüggő és körmentes.
(b) Egy n csúcsú fában $n - 1$ él van.
Indoklás teljes indukcióval:
$n = 2$-re az állítás igaz, mert ekkor a fa két csúcsból és egy darab, a két csúcs közt menő élből áll.
Tegyük fel, hogy igaz az állítás n-re és lássuk be $n + 1$-re, azaz lássuk be, hogy egy $n + 1$ csúcsú F fának n éle van.
A felhasználható állítás szerint az F fában van elsőfokú csúcs, hagyjuk el ezt a csúcsot és a rá illeszkedő élet az F fából. Amit így kapunk az is fa lesz (mert kör nem keletkezhetett ettől és mivel elsőfokú csúcsot hagytunk el, a gráf összefüggő maradt), aminek n csúcsa van. Az indukciós feltevés szerint az új fának $n - 1$ éle van, tehát F-nek n éle volt.

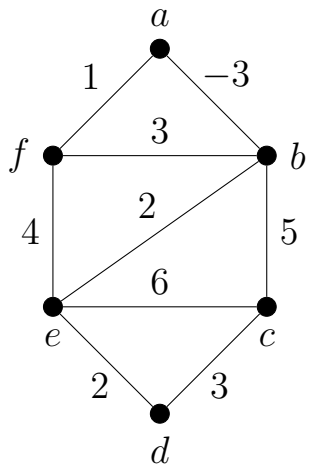
3. Egy irányított gráf élei a következők: $AB, AC, BC, BD, CE, DF, DG, EB, ED, EF$. Futassa le a mélységi bejárást az A csúcsból ezen a gráfon, majd a mélységi bejárás

eredményét felhasználva, az órán tanult módszerrel döntse el, hogy van-e ebben a gráfban topologikus sorrend és ha van, akkor adjon is meg egyet.

Megoldás

Egy lehetséges lefutása a DFS-nek az AB, BC, CE, ED, DF, DG éleket választja be ebben a sorrendben és a csúcsok befejezési számai ezek: $A:7, B:6, C:5, D:3, E:4, F:1, G:2$. A befejezési számok szerinti fordított sorrendet kell vizsgálnunk az órai módszer szerint, azaz az A, B, C, E, D, G, F sorrendet és azt kell megnézni, hogy ebben minden él előre megy-e. Ha igen, akkor ez egy toplogikus sorrend, ha pedig nem, akkor nincs a gráfban topologikus sorrend. Az éleket leellenőrizve látjuk, hogy EB visszafelé megy (B az E előtt van) , azaz ebben a gráfban nincs topologikus sorrend.

4. Az alábbi gráfon futtatjuk Prim algorimusát az a csúcsból, az ab és af élek beválasztása után a *legolcsóbb* tömb így néz ki:



a	b	c	d	e	f
⋆	⋆	5	∞	2	⋆

(a) Hogyan néz ki a *közeli* tömb ekkor és miért?

(b) Melyik élet választja be most az algoritmus és hogyan változik ettől a *legolcsóbb* tömb? Röviden indokolja is a választ.

Megoldás

(a) A *közeli* tömb a már lefedett v csúcsokra azt mutatja, hogy melyik másik csúcsból ide vezető éllel fedtük le v-t, a még nem lefedett v csúcsokra pedig azt mutatja, hogy az eddig lefedett csúcsokból honnan jön a legolcsóbb v-be menő él, ha van egyáltalá ilyen él. Ez alapján a *közeli* tömb így néz ki:

a	b	c	d	e	f
a	*a*	b	∞	b	a

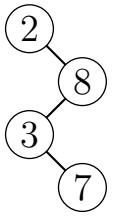
(b) A következő lépésben a *be* élet választjuk, mert az e csúcsnak a legkisebb az értéke a *legolcsóbb* tömbben és *közeli[e] = b*. Ezután a most bekerült e csúcs még nem lefedett w szomszédait kell megnézni (csak itt lehet változás), hogy $c(e,w) <$ *legolcsóbb[w]* fennáll-e. Az e csúcsnak két nem lefedett szomszédja van, c és d, c-nél nem lesz javulás, mert $c(e,c) = 6 >$*legolcsóbb[c] = 5*, de d-nél igen és így *legolcsóbb[d] = 2* lesz.

5. BFS-t (szélességi bejárást) futtatunk az A, B, C, D, E, F csúcsokból álló irányítatlan G gráfon az A kezdőcsúcsból, a BFS fába az AB, AC, BD, CE, CF élek kerülnek be ebben sorrendben. Mely csúcsokkal lehet összekötve az E csúcs a gráfban, melyekkel nem és miért?

   Megoldás Az A csúccsal nem lehet összekötve, mert akkor az A-ból felfedeztük volna fel E-t az első fázisban, azaz lenne AE él a BFS fában.
   A B-vel sem lehet összekötve, mert ha lenne B és E között él, akkor mivel B-t előbb láttuk, mint C-t, ezért B-ből fedeztük volna fel E-t és nem C-ből.
   C-vel össze van kötve, mert van CE él a fában.
   D-vel össze lehet kötve E, mert amikor D szomszédaira sor kerül, akkor már az E be van járva.
   F-fel össze lehet kötve E, mert amikor E szomszédaira sor kerül, akkor már az F csúcs be van járva.

6. Mutassa meg, hogy ha egy 30 csúcsú AVL-fában az 1 és 30 közötti egész számokat tároljuk (az 1 és a 30 is benne van), akkor nem lehetséges, hogy egy érték keresése során a $2, 8, 3, 7$ számokat látjuk ebben a sorrendben.

   Megoldás
   A keresési út alapján tudjuk, hogy a fában van ilyen rész:



   Ez azért van így, mert a 8 a 2 gyereke kell, hogy legyen és a bináris keresőfa tulajdonság miatt tőle jobbra van, és hasonlóan a 3 a 8 bal gyereke, a 7 pedig a 3 jobb gyereke kell hogy legyen.
   Mivel a fában csak egy elem kisebb a 2-nél, ezért a 2-es csúcs bal részfája 1 magas, a jobb fa pedig legalább 3, vagyis itt sérül az AVL-tulajdonság, azaz az, hogy minden csúcsra a csúcs jobb és bal részfájának magassága legfeljebb eggyel térhet el.

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf és adott egy, a csúcsokkal indexelt R tömbben a csúcsok egy részhalmaza oly módon, hogy ha a v csúcs nincsen benne a részhalmazban, akkor $R[v] = 0$, ha pedig v benne van a részhalmazban, akkor $R[v] = 1$.
   Azt szeretnénk eldönteni, hogy igaz-e, hogy mindegyik R-beli csúcspár össze van kötve a gráfban.

   Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

   Megoldás
   Algo: végigmegyünk az R tömbön és ha valahol 1-t látunk egy i csúcsnál, akkor

minden i után szereplő olyan j-re, ahol $R[j]$ is 1 megnézzük, hogy $A[i,j]$ = teljesüle. (Itt A jelöli a gráf szomszédossági mátrixát.) Ha ez minden R-beli 1-es párra teljesül, akkor minden él megvan a gráfban R-beli csúcspárok között, különben meg nem.
Pszeudokódban uez:

```
rendben: = True
ciklus i = 1-tól (n-1)-ig:
    ha R[i] == 1:
        ciklus j = (i + 1)-tól n-ig:
            ha A[i,j] ! = 1:
                rendben := False
        ciklus vége
ciklus vége
```

Jóság: Mivel minden R-beli 1-es párt megnézünk, ezért kiderül, hogy mindegyik benne van-e a gráfban vagy sem.

Lépésszám: A belső ciklus legfeljebb n-szer fut le, a magja konstans sok lépés, vagyis a belső ciklus $O(n)$. A külső ciklus legfeljebb n-szer fut le, ennek magja 1 lépés plusz az $O(n)$-es belső ciklus, azaz a külső ciklus $O(n^2)$-es. Az egész kód csak egy lépéssel több, mint a külső ciklus, azaz ez is $O(n^2)$-es.
Vagy a szöveges leíráshoz: R-ben legfeljebb n darab 1-es lehet és mindegyiknél legfeljebb n másik értéket kell megnéznem R-ben is és A-ban is, azaz legfeljebb $O(n^2)$ lépés történik.

8. Szomszédossági mátrixával adott egy város úthálózatának élsúlyozott, irányított gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak, az élek súlya pedig azt mutatja, hogy mennyi az adott útszakasz hossza. Adott a gráfban két kijelölt él, (u_1, v_1) és (u_2, v_2), és két kijelölt csúcs, A és B, és szeretnénk meghatározni a legrövidebb olyan út hosszát, ami A-ból B-be vezet úgy, hogy legfeljebb az egyik kijelölt élet használja.

   Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk oldani ezt a feladatot? (Itt n szokásos módon a gráf csúcsainak számát jelöli.)

   Megoldás
   Algo: Kitöröljük a gráfból az (u_1, v_1) élet és lefuttatjuk Dijkstra algoritmusát A-ból, majd ugyanezt megtesszük úgy, hogy nem az (u_1, v_1), hanem az (u_2, v_2) élet töröljük. A B csúcsra így kapott két érték közül a kisebbet választjuk.
   A gráf módosítása: a szomszédossági mátrixból az u_i sor v_i oszlapában levő értéket végtelenre állítjuk.

Jóság: Az egyik kitüntetett él törlése után a maradék gráfban éppen azok az utak maradnak, amik legfeljebb a másikat használják. Mindkét lehetőséget figyelembe vettük a két módosítással és Dijkstra használható, mert a probléma természetéből adódóan minden élsúly pozitív.

Lépésszám: a mátrix módosítása 1 lépés, Dijksta $O(n^2)$, azaz a teljes algoritmus $O(n^2) + O(n^2) + O(1) = O(n^2)$.

Algoritmusok és gráfok - Vizsga
2020. január 22.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

1. Az alábbi függvények közül pontosan egyre igaz, hogy $O(n^2)$-es. Válassza ki ezt a függvényt és lássa be megfelelő c konstans és n_0 küszöbérték megadásával, hogy $O(n^2)$-es.

   $$\log(n^2) + \tfrac{n(n-1)(n-2)}{2010} \qquad 2020n\log n + \frac{2^n \cdot n(n-1)}{3^n} \qquad 20n^2(\log n)^2 - 8$$

   Megoldás

   A három függvény közül az $2020n\log n + \dfrac{2^n \cdot n(n-1)}{3^n}$-re igaz, hogy $O(n^2)$-es.
   Ezt úgy látjuk be, hogy megadunk olyan c konstans és n_0 küszöbértéket, melyre $2020n\log n + \dfrac{2^n \cdot n(n-1)}{3^n} \leq cn^2$ fennáll, ha $n \geq n_0$.
   Ezen c és n_0 párt egy egyenlőtlenségrendszerrel találjuk meg:
   $2020n\log n + \dfrac{2^n \cdot n(n-1)}{3^n} \leq 2020n^2 + \dfrac{2^n \cdot n(n-1)}{3^n}$, ha $n \geq 1$, mivel $\log n \leq n$, ha $n \geq 1$.
   $2020n^2 + \dfrac{2^n \cdot n(n-1)}{3^n} \leq 2020n^2 + n(n-1)$, ha $n \geq 1$, mivel $\dfrac{2^n}{3^n} \leq 1$, ha $n \geq 1$.
   $2020n^2 + n(n-1) \leq 2020n^2 + n^2 = 2021n^2$, ha $n \geq 1$, mivel $n - 1 \leq n$, ha $n \geq 1$.
   Azt kaptuk tehát, hogy $2020n\log n + \dfrac{2^n \cdot n(n-1)}{3^n} \leq 2021n^2$, ha $n \geq 1$, azaz $n_0 = 1$ és $c = 2021$ jó választás.

2. A Prim algoritmus lényegi része egy amíg-ciklus, mely két részlet (a két üresen hagyott téglalap) kivételével így néz ki:

```
ciklus amíg van olyan csúcs, ahol legolcsóbb[v] nem *:
    v* := az a csúcs, ahol legolcsóbb[v] minimális
    v* LEFEDVE-be megy
    legolcsóbb[v*] := *
    (közeli[v*], v*) F-be kerül
    ciklus w = 1-tól n-ig:
        ha A[v*,w] nem végtelen és legolcsóbb[w] nem *:
            ha c(v*, w) < legolcsóbb[w]:
                legolcsóbb[w] := [          ]
                közeli[w] := [          ]
    ciklus vége
ciklus vége
```

(a) Mit tárol `legolcsóbb[w]` egy w csúcs esetén?
(b) Mit tárol a `közeli[w]` egy w csúcs esetén?
(c) Egészítse ki a kódot a két üres téglalap kitöltésével és magyarázza el röviden (2-3 mondatban), hogy miért így kell a két hiányzó értéket meghatározni.

Megoldás
(a) legolcsóbb[w] a LEFEDVE halmazból w-be menő legrövidebb él súlyát tárolja
(b) közeli[w] azt a LEFEDVE halmazban levő csúcsot adja meg, ahonnan ez a legrövidebb él jön.
(c) legolcsóbb[w] := c(v*, w) és közeli[w] := v*, mert ha $c(v^*, w) <$ legolcsóbb[w], akkor találtunk egy, az eddigi legjobbnál rövidebb élet w-be a LEFEDVE halmazból, nevezetesen a v^* csúcsból érkező $c(v^*, w)$ súlyú élet.

3. Egy kezdetben üres, 11 méretű hash táblába nyílt címzéssel, lineáris próbával szúrtunk be hét egész számot, a használt hash függvény a $h(K) =$ K maradéka 11-gyel osztva függvény volt. A beszúrások után egy számot kitöröltünk a táblából, a törölt elem helyét * jelzi, az eljárás végén az alábbi táblát kaptuk:

0	1	2	3	4	5	6	7	8	9	10
		14	3		5	17	*	8	20	

(a) Szúrja be ebbe a táblába a 31-et, majd törölje ki a 17-et. Adja meg a tábla állapotát mindkét művelet után és mindkét esetben jelezze, hogy melyik cellákat érintettük.
(b) Mely cellákat érintjük a 19 keresése során az (a) pontban kapott táblában?

Megoldás
(a) A 31-et a $h(31) = 9$-es cellába akarjuk először berakni, de ez foglalt, ezért megyünk balra. A 8-as is foglalt, innen is balra lépünk, a 7-es törölt, ide beszúrhatjuk.
A tábla ez lesz:

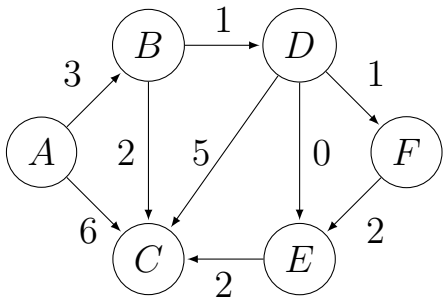
0	1	2	3	4	5	6	7	8	9	10
		14	3		5	17	31	8	20	

A 17-es törlése a 17-es keresésével kezdődik, a $h(17) = 6$-os cellában, itt meg is találjuk, kitöröljük (azaz $*$-ot írunk a helyére.)
A tábla ez lesz:

0	1	2	3	4	5	6	7	8	9	10
		14	3		5	*	31	8	20	

(b) A 19-es érték keresése a $h(19) = 8$-as cellában kezdődik, ez foglalt, megyünk balra, amíg üres cellát nem találunk (azaz átmegyünk a 8, 7, 6, 5 indexű cellákon és végül a 4-es cellában állunk meg, itt látjuk meg, hogy a 19 nincs a táblában.

4. Az alábbi gráfon futtatjuk Dijkstra algorimusát az A csúcsból, az A és B csúcsok KÉSZ-be kerülése után a d tömb így néz ki:



A	B	C	D	E	F
$\star$	$\star$	5	4	∞	∞

(a) Magyarázza el röviden, hogy $d(F)$ miért végtelen és $d(D)$ miért 4.
(b) Melyik csúcs kerül a következő lépésben a KÉSZ halmazba és miért?
(c) Hogyan néz ki a d tömb az új csúcs KÉSZ-be kerülése miatti módosítások után és miért?

Megoldás
(a) A d tömbben $d[v]$ a legrövidebb olyan út hosszát jelöli, ami a kezdőcsúcsból indul, v-be érkezik és az utolsó él kivételével végig a KÉSZ halmazban halad. Mivel F-re nincs ilyen út (mert a KÉSZ halmazból nem vezet él F-be), ezért $d(F) = \infty$.
A D csúcsba csak B-ből érkezhetünk a KÉSZ-ből, a legrövidebb ilyen út az ABD, ennek hossza 4, ezért $d(D) = 4$.
(b) A KÉSZ-be a D kerül be, mert ennek d értéke a legkisebb.
(c) Ha D KÉSZ-be kerül, akkor $d(D) = \star$ lesz és változások lehetnek még D KÉSZ-en kívüli szomszédainál, máshol biztosan nem. D szomszédai C, E, F, ezek mind KÉSZ-en kívül vannak, azt kell megnézni, hogy a D-n át egy éllel ezekbe a csúcsokba vezető legrövidebb út hossza kisebb-e az aktuális d értéknél:
C-re a legjobb D-ből érkező út 9, ez rosszabb, mint az eddigi 5, marad az 5,
E-re és F-re eddig nem volt út, most lett 4 és 5 hosszú, ezért $d(E) = 4, d(F) = 5$.

5. Az $5, 2, x, 10$ tömb összefésüléses rendezése során 4 összehasonlítás történik. Adja meg x összes lehetséges értékét, ha tudjuk, hogy x olyan egész szám, ami máshol nem szerepel a tömbben. Válaszát indokolja is.

Megoldás
Az algoritmus először egy-egy összehasonlítással rendezi az $5, 2$ és $x, 10$ kételemű részeket. Ez eddig biztosan két összehasonlítás, az a kérdés, hogy mikor, mely x értékek esetén lehet két további összehasonlítással befejezni a rendezést.

Megvizsgáljuk az összes esetet:
ha $x < 2$: ekkor a $2, 5$ tömböt fésüljük össze az $x, 10$ tömbbel, de ekkor 2 és x, 2 és 10 , majd 5 és 10 is össze lesz hasonlítva, ez már túl sok összehasonlítás.

Ha $2 < x < 5$, akkor megint csak a $2, 5$ tömböt fésüljük össze az $x, 10$ tömbbel, ekkor 2 és x, 5 és x , majd 5 és 10 is össze lesz hasonlítva, ez ismét túl sok összehasonlítás.
Ha $5 < x < 10$, akkor megint a $2, 5$ és $x, 10$ tömböket fésüljük össze, ekkor 2 és x, majd 5 és x lesz hasonlítva, majd az első tömb elfogy, nem kell több összehasonlítás. Ekkor 4 összehasonlítás volt összesen, ez az eset lehetséges.
Ha $10 < x$, akkor a $2, 5$ és $10, x$ tömböket fésüljük össze, ekkor 2 és 10, majd 5 és 10 lesz hasonlítva, majd az első tömb elfogy, nem kell több összehasonlítás. Ekkor is 4 összehasonlítás volt összesen, ez az eset is lehetséges.

6. Egy G irányított, élsúlyozott gráfban az a, b, c, d, e sorrend topologikus sorrend. Ezen topologikus sorrendet használva a tanult algoritmust futtatjuk az a-tól vett legrövidebb utak hosszának meghatározására. Az a, b, c, d csúcsokra ezeket a távolságokat kaptuk: *távolság*[a] = 0, *távolság[b]* = 2, *távolság[c]* = -7, *távolság[d]* = 5.
(a) Miért biztos, hogy van a gráfban ab él? Mi ennek a súlya és miért?
(b) Ha van a gráfban cd él, akkor mi lehet ennek a súlya és miért?

Megoldás
(a) A b csúcs előtt csak az a van a topologikus sorrendben, ezért egy a-ból b-be menő út csak egy közvetlen a-ból b-be menő él lehet. Mivel ennek az útnak a hossza 2, de az út maga az él, ezért az él hossza 2.
(b) A d-re kapott távolság az algoritmus szerint a legkisebb érték a távolság[a] + c(a,d), távolság[b] + c(b,d), távolság[c] + c(c,d) értékek közül (amennyiben mindhárom él létezik). Ha van él c-ből d-be, akkor távolság[c] $+c(c, d) = -7 + c(c, d) \geq$ távolság[d] = 5, azaz $c(c, d) - 7 \geq 5$, vagyis $c(c, d) \geq 12$.

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf. Adjon $O(n^2)$ lépésszámú algoritmust, ami eldönti, hogy van-e a gráfban olyan másodfokú csúcs, aminek a két szomszédja között van él a gráfban.

Megoldás
Algo szövegesen:
Az A szomszédossági mátrix minden sorára megcsináljuk ezt: végigmegyünk a soron és megszámoljuk, hogy hány 1 van, közben a szomszédos csúcsokat egy listába rakjuk. Ha nem két 1 van, akkor megyünk a következő sorra, ha pedig két darab 1 van, akkor a listában két csúcs van, i és j és megnézzük, hogy $A[i, j]$ 1 vagy 0. Ha 1, akkor találtunk jó másodfokú csúcsot, ha pedig 0, akkor megyünk tovább a következő sorra. Ha végigmentünk az összes soron és így nem találtunk jó másodfokú csúcsot, akkor nincs ilyen.

Algo kóddal:

```
van_jo_csucs := False
ciklus k = 1-tól n-ig:   // végig a sorokon
    szomszedok_szama: = 0
    szomszedok_listaja: =  üres lista
    ciklus s = 1-tol n-ig:            // végig az oszlopokon a k. sorban
        ha A[k,s] == 1:
            szomszedok_szama +=1
            s bele szomszedok_listaja-ba
    ciklus vége
    ha szomszedok_szama == 2:
        ha A[szomszedok_listaja[0], szomszedok_listaja[1]] == 1:
            van_jo_csucs: = True
ciklus vége
return van_jo_csucs
```

Jóság: Az 1-ek száma a sorban a csúcs fokszámát adja meg, pont ezt nézzük meg minden sorban. Ha a szomszédok száma 2, akkor $A[i, j]$ mutatja, hogy ők össze vannak-e kötve vagy sem.

Lépésszám:
A szöveges leíráshoz: Összesen n darab sor van és minden sorban megnézek n értéket, amikor a fokszámokat számolom, értékenként konstans lépés történik (számláló és lista karbantartása), ez eddig $n \cdot O(n) = O(n^2)$. Minden sorban van még egy ellenőrzés (hogy 2 szomszéd volt-e) és legfeljebb egy érték kiolvasása a szomszédossági mátrixból, ez soronként konstans lépés, azaz $O(n)$ összesen, vagyis a lépésszám $O(n^2) + O(n) = O(n^2)$.

Pszeudokód: A külső ciklus n-szer fut le, ennek magja konstans sok lépés, aztán egy belső ciklus, majd konstans sok lépés. A belső ciklus n-szer fut le, magja konstans, vagyis a belső ciklus $O(n)$, a külső ciklus pedig $n(O(1) + O(n) + O(1)) = nO(n) = O(n^2)$. A külső cikluson kívül kívül csak konstans sok lépés van, így az egész kód $O(n^2)$.

8. Szomszédossági mátrixával adott egy n csúcsú, irányított G gráf és benne egy kijelölt (u, v) irányított él. Szeretnénk megtalálni a legkevesebb élből álló, az (u, v) élen átmenő irányított kört a gráfban.

   Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk oldani ezt a feladatot?

   Megoldás
   Algo:

Futtassuk a BFS-t a v csúcsból és nézzük meg a BFS fában a v-ből u-ba vezető utat. Ha nincs ilyen út, akkor nincs ezen az élen át kör, ha van ilyen út, akkor ez az út és az él maga alkotja a legkevesebb élű keresett kört.
Jóság:
Minden uv-n átmenő kör az uv élből és egy vu útból áll, ha megtaláljuk a legrövidebb vu utat, akkor megvan a a legrövidebb keresett kör is.
Lépésszám: BFS lépésszáma (útmegtalálással együtt is) $O(n^2)$, ezután a vu út megkeresése a fában $O(n)$, azaz együtt $O(n^2)$.

Algoritmusok és gráfok - Vizsga
2020. január 29.

A válaszokat indokolni kell, de a feladatokban szereplő tanult algoritmusokat nem kell részletesen leírni, elég csak azokat a részeket kifejteni, amelyek az indokláshoz szükségesek.

1. Az alábbi pszeudokód inputja egy $n \geq 2$ hosszú A tömb. Igaz-e, hogy ez a kód $O(n)$-es, ha egy lépésnek az összehasonlítás, az összeadás és az értékadás számít? Ha igaz, akkor lássa be ezt, ha pedig nem igaz, akkor lássa be, hogy $O(n^2)$-es.

```
ciklus i = 0-tól (n-1)-ig:
    ha A[i] > 0:
        ciklus amíg A[i] < 5:
            A[i] := A[i] + 1
        ciklus vége
ciklus vége
```

Megoldás
Ez a kód $O(n)$-es. A külső ciklus n-szer fut le, ennek a magja pedig $O(1)$-es a következők miatt: Egy lépés után egy belső ciklus következik, ami legfeljebb 5-ször fut le, mert minden lefutáskor nő eggyel az $A[i]$ értéke és mivel legalább 0 volt, ezért legfeljebb 5 lefutás után nagyobb lesz, mint 5 és akkor már nem teljesül az amíg-feltétel. A belső ciklus magja konstans sok lépés, így a belső ciklus $O(1)$ lépésből áll, azaz a külső ciklus magja $1 + O(1)$, azaz $O(1)$, így az egész kód $O(n)$.

2. (a) Adja meg a bináris keresőfa definícióját (elég a bináris keresőfa tulajdonságot leírnia, nem kell definiálni azt, hogy mi a bináris fa).
(b) Írja le, hogy hogyan kell beszúrni egy új elemet bináris keresőfába. Mennyi ennek az eljárásnak a lépésszáma és miért?

Megoldás
(a) A bináris keresőfa tulajdonság azt jelenti, hogy minden csúcsra igaz, hogy a bal részfájában minden elem kisebb, a jobb részfájában pedig minden elem nagyobb nála.
(b) Egy új s érték beszúrása úgy történik, hogy először s-et összehasonlítjuk a gyökérrel és ha s kisebb, mint a gyökér, akkor balra, ha pedig nagyobb, akkor jobbra megyünk. Ezt az eljárást folytatjuk az aktuális csúccsal (ahova érkezünk) mindaddig, amíg üres gyerekhez (None mutatóhoz) érünk, ekkor ide beillesztjük az s értéket.
Ezen eljárás lépésszáma $O(h)$, ahol h a fa magassága, mert minden szinten konstans sok lépést teszünk.

3. Egy irányított gráf élei a következők: $AB, AC, BG, CB, CD, CG, DA, DE, DF, EF, GD, GF$. Ebben a gráfban BFS-t (szélességi bejárást) futtatunk az A kezdőcsúcsból úgy, hogy ha bármikor választási lehetőség adódik, akkor az ábécé szerint előbb levő csúcsot választjuk.
(a) Milyen sorrendben járjuk be a csúcsokat és mi lesz a kapott BFS fa? (Indokolni

ezt nem kell, elég a sorrend és a fa megadása.)
(b) Hogyan néz ki a *honnan* tömb a BFS futásának a végén? Indoklásképpen írja le röviden, hogy mit tárol a *honnan* tömb.
(c) Adja meg a Q sor állapotát a kód futásának kezdetén, majd minden egyes csúcs bejárása után. Indoklásképpen röviden írja le, hogy a Q sornak mi a szerepe a kódban.

Megoldás
(a) A, B, C, G, D, F, E, a fa élei pedig: AB, AC, BG, CD, GF, DE.
(b)

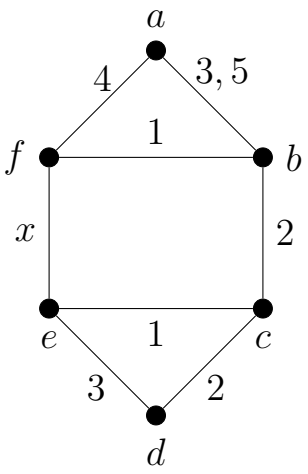
A	B	C	D	E	F	G
A	A	A	C	D	G	B

Ez azért van így, mert a $honnan[v]$ azt tárolja, hogy a BFS eljárás során a v csúcsot melyik csúcsból fedeztük fel.
(c) A Q sor azokat a csúcsokat tárolja, amiket már felfedeztünk, de amiknek a szomszédait még nem vizsgáltuk meg.
Az elején $Q = \{A\}$, később pedig
A bejárása után $Q = \{B, C\}$,
B bejárása után $Q = \{C, G\}$,
C bejárása után $Q = \{G, D\}$,
G bejárása után $Q = \{D, F\}$,
D bejárása után $Q = \{F, E\}$,
F bejárása után $Q = \{E\}$,
E bejárása után $Q = \{\}$.

4. A G irányított, élsúlyozott gráf élei a következők (zárójelben az élsúlyok):
$ac(8), af(7), bf(11), cd(10), ce(7), cf(6), eb(-5), ed(2), ef(3)$.
Ebben a gráfban az órán tanult, DAG-ban leghosszabb utat kereső algoritmust futtatjuk a-ból az a, c, e, b, d, f topologikus sorrendet használva. Az a, c, e, b, d csúcsokba vezető leghosszabb utak hosszát már meghatároztuk: *leghosszabb[a] = 0*, *leghosszabb[c] = 8*, *leghosszabb[e] = 15*, *leghosszabb[b] = 10*, *leghosszabb[d] = 18*. Hogyan kapjuk meg *leghosszabb[f]*-et a tanult eljárással ezekből az adatokból? A megoldásából látszódjon, hogy hogyan használja a tanult eljárást.

Megoldás
Az összes olyan v csúcsra, ahonnan vezet él f be, meg kell néznünk a *leghosszabb[v] + c(v,f)* értéket és ezek közül kell vennünk a maximumot. Ez azt jelenti, hogy a maximumát keressünk a
leghosszabb[a] + c(a,f), leghosszabb[c] + c(c,f), leghosszabb[e] + c(e,f),
leghosszabb[b] + c(b,f) értékeknek, ami b-nél található 21 értékben, azaz
leghosszabb[f] értéke 21.

5. Az alábbi gráfon, ahol az x élsúly nem ismert, futtatva Kruskal algorimusát az első három él, amit az eljárás beválaszt: bf, ec, bc. Adja meg x összes lehetséges értékét (röviden indokolva válaszát) és írja le, hogy mely éleket és milyen sorrendben választja be ezután az algoritmus és miért.



Megoldás

Az x értékére az igaz, hogy $x \geq 2$, mert a Kruskal algoritmus sorbarendezi az éleket súlyt szerint és az aktuális éle akkor veszi be a fába, ha az nem alkot kört az eddig választott élekkel (egyébként átlépi és megy a következő él vizsgálatára).
H x kisebb lenne, mint 2, akkor az fe élet vizsgálná bf és ec után és mivel nem alkotna velük kört, ezért ezt az fe élet venné be harmadiknak.
A bf, ec, bc élek után az fe élet biztos nem veheti be sose, mert kört alkotna, a többi élet pedig cd, ed, ab, af sorrendben nézi és ezekből először a cd, majd az ab élet veszi be.

6. Egy kezdetben üres, 11 méretű hash táblába nyílt címzéssel, lineáris próbával szúrtunk be nyolc egész számot, a használt hash függvény a $h(K) = \text{K}$ maradéka 11-gyel osztva függvény volt. A beszúrások után két számot kitöröltünk a táblából, a törlések helyét $*$ jelzi. Mutassa meg, hogy miért nem lehetséges, hogy az eljárás végén az alábbi táblát kaptuk:

0	1	2	3	4	5	6	7	8	9	10
16	*		14	6	*	17			9	21

Megoldás

A 16 nem kerülhetett oda, ahol van, mert ezt először a $h(16) = 5$-ös cellába akarjuk rakni és ha ide nem tudjuk, akkor balra indulva keresünk neki új helyet, amit legkésőbb a 2-es cellában megtalálnánk (mert ott sose volt semmi), vagyis ezen a cellán nem juthatott volna túl a 16.

7. Szomszédossági mátrixával adott egy n csúcsú, irányítatlan G gráf és adott egy, a csúcsokkal indexelt R tömbben a csúcsok egy részhalmaza oly módon, hogy ha a v csúcs nincsen benne a részhalmazban, akkor $R[v] = 0$, ha pedig v benne van a részhalmazban, akkor $R[v] = 1$.
Azt szeretnénk eldönteni, hogy igaz-e, hogy egyetlen él sem fut R-beli csúcsok közt a gráfban.

Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

Megoldás

Algo: végigmegyünk az R tömbön és ha valahol 1-t látunk egy i csúcsnál, akkor minden i után szereplő olyan j-re, ahol $R[j]$ is 1 megnézzük, hogy $A[i, j] = 0$ teljesüle. (Itt A jelöli a gráf szomszédossági mátrixát.) Ha ez minden R-beli 1-es párra teljesül ez, akkor egyetlen él sincsen a gráfban R-beli csúcspárok között, különben meg ez nem igaz

Pszeudokódban uez:

```
rendben: = True
ciklus i = 1-tól (n-1)-ig:
    ha R[i] == 1:
        ciklus j = (i + 1)-tól n-ig:
            ha (R[j] == 1 és A[i,j] ! = 0):
                rendben := False
        ciklus vége
ciklus vége
```

Jóság: Mivel minden R-beli 1-es párt megnézünk, ezért kiderül, hogy mindegyik él hiányzik-e a gráfból vagy sem.

Lépésszám: A belső ciklus legfeljebb n-szer fut le, a magja konstans sok lépés, vagyis a belső ciklus $O(n)$. A külső ciklus legfeljebb n-szer fut le, ennek magja 1 lépés plusz az $O(n)$-es belső ciklus, azaz a külső ciklus $O(n^2)$-es. Az egész kód csak egy lépéssel több, mint a külső ciklus, azaz ez is $O(n^2)$-es.

Vagy a szöveges leíráshoz: R-ben legfeljebb n darab 1-es lehet és mindegyiknél legfeljebb n másik értéket kell megnéznem R-ben is és A-ban is, azaz legfeljebb $O(n^2)$ lépés történik.

8. Egy A szomszédossági mátrix-szal adott egy város úthálózatának élsúlyozott, irányított gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak, az élek súlya pedig azt mutatja, hogy mennyi az átlagos idő, ami az út megtételéhez autóval szükséges. Adott továbbá egy másik B mátrix is, ami azt adja meg minden közvetlen útra, hogy mekkora a legnagyobb magasságú autó, amivel elakadás nélkül át lehet haladni az adott szakaszon: a B mátrix i-edik sorának, j-edik oszlopában álló $B[i, j]$ érték mutatja a legnagyobb magasságot, amivel még megtehető az i csomópontból a j csomópontba vezető közvetlen út.

   Adott a gráfban két kijelölt csúcs, X és Y. Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk találni a leggyorsabban megtehető olyan utat X-ből Y-ba, amit legfeljebb 350 centiméter magas autóval meg lehet tenni? (Itt n szokásos módon a gráf csúcsainak számát jelöli.)

Megoldás

Algoritmus:

1. Módosítsuk a gráfot úgy, hogy azokat az éleket, amiken nem lehet átmenni 350 cm magas autóval, azokat kitöröljük.
2. Ebben az új gráfban futtassuk a Dijkstra algoritmust az X csúcsból. Ha nem kapunk utat Y-ba, akkor nincs ilyen út, ha kapunk, akkor pedig ez lesz a legrövidebb kívánt út.

A gráf módosítása pontosabban: végigmegyek az A szomszédossági mátrixon, soronként, soron belül oszloponként és ha $A[i,j]$ nem végtelen, akkor megnézem, hogy $B[i,j] \geq 350$ teljesül-e. Ha igen, akkor nincs tennivalónk ezzel a bejegyzéssel, egyébként pedig $A[i,j]$-t végtelenre állítjuk.

```
ciklus 1 =1-tól n-ig:
    ciklus j = 1-tól n-ig:
        ha A[i,j] nem végtelen  és B[i,j] < 350:
            A[i,j] := végtelen
    ciklus vége
ciklus vége
```

Lépésszám:

1. Az új szomszédossági mátrix előállítása során minden cellánál konstans sok munkát végzek, azaz ez $O(n^2)$ lépés. (Vagy a pszeudokódra hivatkozva: a külső ciklus n-szer fut le, ennek a magja a belső ciklus, ami egy n-szer lefutó, konstans magú kód, azaz a belső ciklus $O(n)$, az egész kód pedig $O(n^2)$.)
2. A kapott mátrix is egy n csúcsú gráfhoz tartozik, ebben Dijkstra $O(n^2)$ lépésben fut, így a módosítás és a futtatás együtt $O(n^2) + O(n^2) = O(n^2)$.

Jóság: A módosítással éppen azok az élek válnak elérhetetlenné, amiket nem szabad használnunk, azaz az új gráf útjai megegyeznek az eredeti gráf használható útjaival. Dijkstra algoritmusa használható, mert minden élsúly pozitív.