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.

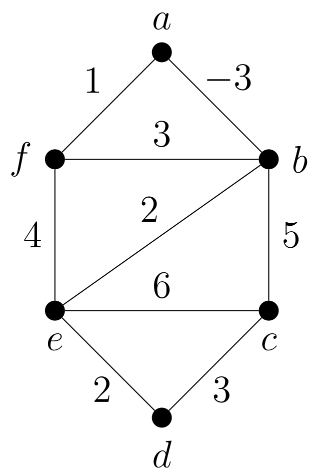
Megoldás
(a) Egy irányítatlan gráf akkor fa, ha összefüggő és körmentes.
(b) Egy n csúcsú fában $n-1$ él van.
Indoklás teljes indukcióval:
$n = 2$-re az állítás igaz, mert ekkor a fa két csúcsból és egy darab, a két csúcs közt menő élből áll.
Tegyük fel, hogy igaz az állítás n-re és lássuk be $n+1$-re, 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
$\star$	$\star$	5	∞	2	$\star$

(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

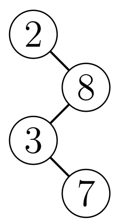
(b) A következő lépésben a be élet választjuk, mert az e csúcsnak a legkisebb az értéke a *legolcsóbb* tömbben és *közeli[e]* $= b$. Ezután a most bekerült e csúcs még nem lefedett w szomszédait kell megnézni (csak itt lehet változás), hogy $c(e, w) <$ *legolcsóbb[w]* fennáll-e. Az e csúcsnak két nem 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 $A[i,j] =$ teljesül-e. (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 $O(n^2) + O(n^2) + O(1) = O(n^2)$.