Algoritmusok és gráfok
TIZEDIK GYAKORLAT, 2019. november 22.
MEGOLDÁSOK

1. (a) Rajzolja fel az alábbi (élsúlyozott) szomszédossági mátrixhoz tartozó irányított gráfot (a csúcsok sorban legyen a, b, c, d, e, f).
(b) A lerajzolt gráfot alaposan megnézve lássa be, hogy ez a gráf nem DAG.
(c) Az órán tanult, mélységi bejárást használó eljárással is lássa be, hogy ez nem egy DAG.
(d) Hogyan módosul a mátrix, ha az fc él helyett cf él van a gráfban?
(e) Az órán tanult módszerrel keressen egy topologikus sorrendet ebben a módosított gráfban.

$$\begin{bmatrix} \infty & 3 & 2 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 2 & \infty & -6 & \infty & \infty \\ \infty & \infty & \infty & \infty & 10 & 0 \\ \infty & 8 & \infty & \infty & \infty & \infty \\ \infty & 3 & 7 & \infty & \infty & \infty \end{bmatrix}$$

Megoldás

(b) Látható, hogy a cdf egy irányított kör.
(c) Ha az a csúcsból lefutttajuk a mélységi bejárást, akkor a befejezési számok szerinti fordított sorrend a, c, d, f, e, b. Végignézve a gráf éleit láthatjuk, hogy az fc él hátrafelé vezet, azaz ez nem topologikus sorrend, de akkor a tanult tétel szerint a gráf nem volt DAG.
(d) Az utolsó sor 3. elem végtelen lesz, a 3. sor utolsó elem meg 7.
(e) Lefuttatjuk a mélységi bejárást bárhonnan, pl. f-ből, majd újrakezdve a-ból, a befejezési számok szerinti fordított sorrend a, c, d, e, f, b lesz, amiben a gráf minden éle előre megy, azaz ez egy toplogikus sorrend.

2. Az irányított G_1 gráf csúcsai a, b, c, d, e, élei pedig ab, ac, bd, cd, de, ea, az irányított G_2 gráf csúcsai a, b, c, d, e, f, g, élei pedig $ag, af, ba, bg, cd, ed, fe, fc, fd, gf, ge$.
Használja mindkét gráfra az órán tanult, mélységi bejárást használó módszert és vagy lássa be vele, hogy a gráf nem DAG vagy adjon meg benne egy toplogikus sorrendet.

Megoldás

(a) Ha az a csúcsból lefutttajuk a mélységi bejárást, akkor a befejezési számok szerinti fordított sorrend a, c, b, d, e. Végignézve a gráf éleit láthatjuk, hogy az ea él hátrafelé vezet, azaz ez nem topologikus sorrend, de akkor a tanult tétel szerint a gráf nem volt DAG és akkor nincs is benne toplogikus sorrend.
(b) Lefuttatjuk a mélységi bejárást bárhonnan, pl. a-ból, a befejezési számok szerinti fordított sorrend b, a, g, f, e, c, d lesz, amiben a gráf minden éle előre megy, azaz ez egy topologikus sorrend és a gráf DGA.

3. **(nagyjából Vizsga 2019)** Igaz-e, hogy az alábbi irányított, élsúlyozott gráfban a B, C, D, A, E, F sorrend egy topologikus sorrend? A gráf (irányított) élei: $AE, AF, BC, BD, BE, CA, CE, DE, DF$

Megoldás

Csak végig kell nézni, hogy minden él előre megy-e és ez igaz, vagyis ez egy toplogikus sorrend.

4. Egy mátrixával adott irányított G gráfban néhány csúcs piros, néhány csúcs kék, a többi csúcs pedig színtelen. (A színezés egy, a csúcsokkal indexelt C tömbben adott). Melyik tanult algoritmus egyszeri futásával lehet $O(n^2)$ lépésben megoldani az alábbi feladatokat? Mindegyik esetben írja le, hogy hogyan kell módosítani az inputot (ha kell), melyik algoritmust futtatjuk és hogyan és a futás

után hogyan kapjuk meg a választ a kérdésre.
(a) El akarjuk dönteni, hogy az egyik adott p_1 piros csúcsból mik az elérhető kék csúcsok. (Elérhető azt jelenti, hogy van oda irányított út.)
(b) Adott két piros csúcs, p_1 és p_2 és el akarjuk dönteni, hogy mely kék csúcsok érhetőek el legalább az egyikükből.
(c) El akarjuk dönteni, hogy mindegyik kék csúcs elérhető-e legalább egy pirosból (azaz igaz-e, hogy minden kék csúcsra van olyan piros csúcs ahonnan ő elérhető).
(d) El akarjuk dönteni egy adott k_1 kék csúcsról, hogy ő mely piros csúcsokból érhető el.
(e) Egy adott piros p_1 csúcshoz meg akarjuk keresni a legközelebbi kék csúcso(ka)t.(Legközelebbi azt jelenti, hogy legkevesebb élből álló úton érjük el.)
(f) Meg akarjuk keresni a legrövidebb olyan utat a gráfban, ami piros csúcsból kék csúcsba vezet.

Megoldás az (a)- részre, nem teljes megoldások, csak vázlatok a többire

(a) Szélességi vagy mélységi bejárást futtatunk p_1-ből (nem kell módosítani a mátrixot), majd megnézzük, hogy mely csúcsokra igaz, hogy kékek és be vannak járva a végén. Ezen utóbbi lépés pszeudokódja (inputja a bejárás végén kapott *bejárva* tömb és a színeket tartalmazó C tömb, outputja a *bejárt_kék* tömb):

```
bejárt_kék[v] : = 0 minden v-re
ciklus i = 1-től n-ig:
    ha C[i] == 1 és bejárva[i] == 1:
        bejárva_kék[i] := 1
ciklus vége
```

Ez utóbbi kód $O(n)$-es, mert egy n-szer lefutó konstans magú ciklusról van szó, így az egész eljárás $O(n^2)$.

A megoldás azért helyes, mert mindkét bejárás pont azokat a csúcsokat járja be, amik elérhetőek a kezdőcsúcsból.

(b) Ötlet: adjunk hozzá egy új csúcsot a gáfhoz, amiből vezessünk egy élet a két kitüntetett piros csúcsba. Ebben az új gráfban nézzük meg valamelyik bejárással, hogy melyik kékek érhetők el az új csúcsból, pont ezek lesznek elérhetők valamelyik pirosból a kettő közül.

Jóság: Ha van út p_1-ből vagy p_2-ből egy kékbe az eredeti gráfban, akkor az újban lesz út az új csúcsból ide és ha van út az új gráfban egy kék csúcsba az új csúcsból, akkor annak első éle biztosan vagy p_1-be vagy p_2-be vezetett (mert csak ide van él az új csúcsból), vagyis az út maradék része p_1-ből vagy p_2-ből vezet a kék csúcsba.

Algo pontosabban:

Az új mátrix létrehozása az eredeti A mátrixból: felveszünk egy (n+1)-szer (n+1)-es csupa nulla mátrixot, ennek neve B, ebben az utolsó sor/oszlop lesz az új csúcs (ez $O(n+1)^2$, azaz $O(n^2)$ lépés), majd

```
ciklus i = 1-től n-ig:
    ciklus j = 1-től n-ig:
        B[i,j] : = A[i,j]    \\ átmásolom az eredeti csúcsok közti éleket
    ciklus vége

    B[i,n+1] : = 0  \\ nem vezet él az új csúcsba egik régiből sem
ciklus vége

B[n+1, p_1] : = 1    \\ él az egyik pirosba az újból
B{n+1, p_2] : = 1    \\ él a másik pirosba az újból
```

Ez $O(n^2)$, mert a külső ciklus n-szer fut le, ennek magja egy konstans magú, n-szer lefutó belső ciklus és még 1 lépés, azaz n-szer fut le egy $1 + O(n)$-es, azaz $O(n)$-es mag, vagyis ez eddig $O(n^2)$, ezután pedig csak 2 lépés van, de $O(n^2) + 2$ az $O(n^2)$.

Azt, hogy mely kékek érhetőek el, azt ugyanúgy nézzük meg, mint az (a) részben.

(c) Hasonlóan a (b) részhez, csak itt minden pirosba megy majd él az új csúcsból (a mátrix módosításában a kód utolsó két sora helyett egy ciklus kell majd, amiben beállítjuk, hogy minden piros csúcsbe menjen él az új csúcsból).

(d) Ötlet: Elkészítjük a gráf fordítottját (akkor van él az újban u-ból v-be, ha az eredetiben volt v-ből u-ba), ebben futtatjuk valamelyik bejárást a kijelölt kék csúcsból.

(e) Szélességi bejárást futtatunk p_1-ből, azt a változatot, amiben távolságot is számolunk és a végén megnézem, hogy a kék csúcsokra mik a távolságok, ezek között keressük a legkisebb értékeket.

(f) Új kezdőcsúcsot veszünk fel, ebből megy él minden pirosba és ebből az új csúcsból csináljuk azt, amit az (e) részben p_1-ből.