Algoritmusok és gráfok
KILENCEDIK GYAKORLAT, 2019. november 13.
MEGOLDÁSOK pár feladathoz

1. Az alábbi szomszédossági mátrix-szal adott G irányított gráfot járja be mélységi bejárással az első csúcsból.
(a) Milyen sorrendben érjük el a csúcsokat?
(b) Mik a csúcsok befejezési számai?
(c) Hogyan néz ki a futás végén a *honnan* tömb?
(d) Hogyan ágyazódnak egymásba a $DFS(G,v)$ függvényhívások?

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Megoldás Nevezzük el a csúcsokat sorrendben a, b, c, d, e, f, g, h-nak.
(a) A csúcsokat a, b, c, d, f, g, h, e sorrendben érjük el.
(b) A befejezési sorrend: b, f, h, g, d, e, c, a.
(c) A *honnan* tömbben: $honnan[a] = a$, $honnan[b] = a$, $honnan[c] = a$, $honnan[d] = c$, $honnan[e] = c$, $honnan[f] = d$, $honnan[g] = d$, $honnan[h] = g$
(d) A függvényhívások így ágyazódnak egymásba: $DFS(G,a)$ meghívja $DFS(G,b)$-t és $DFS(G,c)$-t. $DFS(G,b)$ nem hív meg senkit. $DFS(G,c)$ meghívja $DFS(G,d)$-t és $DFS(G,e)$-t, $DFS(G,d)$ meghívja $DFS(G,f)$-et és $DFS(G,g)$-t, $DFS(G,e)$ nem hív meg senkit, $DFS(G,f)$ nem hív meg senkit, $DFS(G,g)$ meghívja $DFS(G,h)$-t, $DFS(G,h)$ nem hív meg senkit.

2. Egy város úthálózata szomszédossági mátrixával adott irányítatlan gráffal van leírva, a csúcsok a kereszteződések, az élek pedig a kereszteződések közt vezető utak. Filmforgatás miatt néhány utcát lezárnak, tudjuk, hogy melyeket, ez az információ egy másik nxn-es L mátrixban van megadva úgy, hogy $L[i,j] = 1$, ha az i és j csomópont között lezárás van, egyébként $L[i,j] = 0$. Adjon $O(n^2)$ lépésszámú algoritmust, ami a szomszédossági mátrix módosításával és egy tanult algoritmus változtatás nélküli futtatásaival eldönti, hogy el tudunk-e jutni otthonunkból (ami egy csúcspontja a gráfnak) az egyetemre (ami egy másik csúcsa a gráfnak) a felszínen, csak létező lezáratlan utakat használva.

Megoldás

Ötlet: Módosítsuk a szomszédossági mátrixot úgy, hogy a lezárt utaknak megfelelő éleket töröljük belőle és az új gráfban futtassuk a szélességi vagy a mélységi bejárást az otthonunkhoz tartozó csúcsból. Ha a bejárás végén elérhető az egyetemi csúcs, akkor el tudunk jutni, különben meg nem.

Pontosabban a mátrix módosítása: A szomszédossági mátrixon végigmegyünk soronként, soron belül balról jobbra és ha valahol 1 van (azaz ott van él), akkor megnézzük az L kétdimenziós tömbben, hogy ez le van-e zárva. Ha igen, akkor átírjuk a szomszédossági mátrixot itt 0-ra.

Pszeudokódban ugyanez:

```
ciklus i =1-től n-ig:
    ciklus j= 1-től n-ig:
        ha A[i,j] == 1 és L[i,j] == 1:
            A[i,j] := 0
    ciklus vége
ciklus vége
```

Ezután a szélességi bejárást futtatjuk az új mátrixon az otthonunkhoz tartozó csúcsból (ezt már nem kell leíni részletesen, elég azt megmondani, hogy pontosan mely gráfon (mátrixon) és honnan futtatjuk).

Lépésszám: A mátrix módosítása $O(n^2)$, mert a külső ciklus n-szer fut le, ennek a magja egy másik ciklus, ami n-szer fut le és magha $O(1)$, vagyis a külső ciklus magja $O(n)$, az egész módosítás meg $O(n^2)$. Ezután egy n csúcsú gráfon fut a BFS vagy DFS, ami $O(n^2)$ (ezt tudjuk óráról), utána meg csak azt kell megnézni 1 lépésben, hogy az egyetem csúcsa be van-e járva. Vagyis az egész $O(n^2) + O(n^2) + 1$, vagyis $O(n^2)$.

Jóság: A módosítás utáni gráfban pontosan akkor érhető el az egyetem otthonról, ha a lezárások ellenére is el lehet otthonról az egyetemre jutni, azt meg tanultuk, hogy a BFS és a DFS el tudja dönteni, hogy egy gráfban mely csúcsok elérhetők a kezdőcsúcsból.

3. Egy hat pontú irányított gráf csúcsait egy mélységi bejárás a, c, f, e, d, b sorrendben járja be, a befejezési számok pedig ezek: a: 6; b: 5; c: 4; d: 3; e: 2; f: 1.
(a) Lehetséges-e, hogy a gráfban van él f-ből e-be?
(b) Lehetséges-e, hogy a gráfban van él d-ből e-be?

Megoldás

A felfedező élek ac, cf, ce, cd, ab, mert

- a-ból c-be megyünk először, de c-ben nem fejezzük be (mert nem ő az első, akit befejezünk), ezért az f csúcsot c-ből tovább menve kell látnunk.
- f-ből itt visszafordultunk c-be (különben egy másik csúcsnak lenne 1 a befejezési száma) és mivel c befejezési száma nem 2, ezért a következő csúcsot, e-t c-ből látogatjuk meg, de e befejezési száma 2, vagyis innen már nem megyünk tovább, visszalépünk c-be
- mivel c befejezési száma nem 3, ezért a következő csúcsot, d-t is c-ből látogatjuk meg, de d befejezési száma 3, vagyis innen már nem megyünk tovább, visszalépünk ismét c-be
- c befejezési száma 4, vagyis innen most visszafordulunk a-ba, amit még nem fejezünk be, vagyis a innen látjuk b-t, az utolsó csúcsot.

(a) Biztos, hogy nincs él f-ből e-be, mert akkor f-et nem fejeztük volna be elsőnek, hanem tovább léptünk volna belőle e-be.
(b) Lehet él d-ből e-be, mert amikor d szomszédait nézzük, akkor e már be van járva, mindegy, hogy van-e él d-ből e-be.

4. Egy mátrixával adott irányítatlan G gráfban minden csúcs ki van színezve, piros, zöld vagy kék színre (ez az információ egy, a csúcsokkal indexelt C tömbben adott). Adott egy piros s és egy piros t csúcs, szeretnénk meghatározni az s-ből t-be vezető legrövidebb olyan út hosszát, ami csak piros csúcsokon megy át. Adjon erre a feladatra $O(n^2)$ lépésszámú algoritmust.

Megoldás

Ötlet: Módosítsuk a szomszédossági mátrixot úgy, hogy csak piros csúcsok közti élek maradjanak, majd futtassuk a szélességi bejárás távolságot számoló változatát.

Pontosabban a mátrix módosítása: A szomszédossági mátrixon végigmegyünk soronként, soron belül balról jobbra és ha valahol 1 van (azaz ott van él), akkor ott megnézzük, hogy mindkét csúcs piros-e és ha nem, akkor átírjuk a szomszédossági mátrixot itt 0-ra.

Pszeudokódban ugyanez:

```
ciklus i =1-tól n-ig:
    ciklus j= 1-től n-ig:
        ha A[i,j] == 1:
            ha (C[i] != piros vagy  C[j] != piros):
                A[i,j] := 0
    ciklus vége
ciklus vége
```

Ezután a szélességi bejárást futtatjuk az új mátrixon az s csúcsból és megnézzük *távolság[t]*-t.

Lépésszám: A mátrix módosítása $O(n^2)$, mert a külső ciklus n-szer fut le, ennek a magja egy másik ciklus, ami n-szer fut le és magha $O(1)$, vagyis a külső ciklus magja $O(n)$, az egész módosítás meg $O(n^2)$. Ezután egy n csúcsú gráfon fut a BFS, ami $O(n^2)$ (ezt tudjuk óráról), utána meg csak azt kell megnézni 1 lépésben, hogy mennyi *távolság[t]*. Vagyis az egész $O(n^2) + O(n^2) + 1$, vagyis $O(n^2)$.

Jóság: A módosítás utáni gráfban pontosan azok az utak maradnak meg, amik csak piros csúcsok közti élekből állnak, vagyis az új gráf útjai pontosan a piros utak, azt meg tanultuk, hogy a BFS egy gráfban meghatározza a a távolságokat, vagyis a legrövidebb utak hosszát, de mivel az új gráf útjai az eredeti gráf piros útjai, ezért a legrövidebb piros út hosszát kapjuk meg.

5. (a) Hogyan zajlik az n pontú irányítatlan teljes gráf (ahol minden pont minden másik ponttal össze van kötve) szélességi bejárása? Hogy néz ki a BFS fa?
(b) Hogyan zajlik az n pontú irányítatlan teljes gráf mélységi bejárása? Hogy néz ki a DFS fa?

Megoldás

(a) Egy csillag, azaz a kezdőcsúcsból minden másik csúcsba vezet egy él.
(b) Egy n hosszú út lesz a fa.

6. Egy kezdő autóvezető a városban való közlekedése során szeretne gyakorlatának megfelelő útvonalat választani. Az úthálózat egy irányítatlan gráfként van megadva, a csúcsok a kereszteződések, az élek az utak, a csúcsoknál adott, hogy nehéz-e számára az a kereszteződés. A gráf szomszédossági mátrixával adott, az az információ pedig, hogy mely kereszteződések nehezek, egy, a csúcsokkal indexelt N tömbben adott, ahol $N[v] = 1$, ha a csomópont nehéz, különben $N[v] = 0$.

Adjon $O(n^2)$ lépésszámú algoritmust, amivel meg lehet határozni, hogy az autós az egyik adott csúcsnál levő otthonából mely csúcsokba tud autóval úgy eljutni, hogy útja során két nehéz csúcs soha nem jön közvetlenül egymás után.

Megoldás

Ötlet: Módosítsuk a szomszédossági mátrixot úgy, hogy csak akkor maradjon meg egy él, ha legalább az egyik csúcs nem veszélyes, majd futtassuk a szélességi vagy mélységi bejárást az otthonunkból. Azok a csúcsok lesznek biztonságosan elérhetők, amik ebben az új gráfban bejártak.

Pontosabban a mátrix módosítása: A szomszédossági mátrixon végigmegyünk soronként, soron belül balról jobbra és ha valahol 1 van (azaz ott van él), akkor ott megnézzük, hogy mindkét csúcs veszélyes-e és ha igen, akkor átírjuk a szomszédossági mátrixot itt 0-ra.

Pszeudokódban ugyanez:

```
ciklus i =1-től n-ig:
    ciklus j= 1-től n-ig:
        ha A[i,j] == 1:
            ha (N[i] == 1 és  N[j] == 1):
                A[i,j] := 0
    ciklus vége
ciklus vége
```

Ezután a szélességi vagy mélységi bejárást futtatjuk az új mátrixon az otthonunkból csúcsból és megnézzük, hogy kik bejártak a *bejárva* tömbben.

Lépésszám: A mátrix módosítása $O(n^2)$, mert a külső ciklus n-szer fut le, ennek a magja egy másik ciklus, ami n-szer fut le és magha $O(1)$, vagyis a külső ciklus magja $O(n)$, az egész módosítás meg $O(n^2)$. Ezután egy n csúcsú gráfon fut a BFS, ami $O(n^2)$ (ezt tudjuk óráról), utána meg csak azt kell megnézni, hogy ki lett bejárva, ami $O(n)$ (mert a tömbön kell végigmennünk). Vagyis az egész $O(n^2) + O(n^2) + O(n)$, vagyis $O(n^2)$.

Jóság: A módosítás utáni gráfban pontosan azok az utak maradnak meg, amik biztonságosak, vagyis az új gráf útjai pontosan a biztonságos utak, azt meg tanultuk, hogy a mindkét bejárás meghatározza, hogy mely csúcsok elérhetők a kezdőcsúcsból úton.