

A számítástudomány alapjai 2014. I. félév

4. gyakorlat. Összeállította: Fleiner Tamás (fleiner@cs.bme.hu)

Tudnivalók

Def: A $G = (V, E)$ (irányított vagy irányítatlan) gráf egy *bejárásán* a V -beli csúcsok végiglátogatását értjük, ahol a alábbiak szerint. Az újonnan elért csúcsot ha lehetséges mindig már korábban elért csúcsból kell egy oda vezető él mentén elérni. Ha ez nem lehetséges, és még van eléretlen csúcs, akkor tetszőleges csúccsal lehet a következőnek elért csúcs. (Irányítatlan gráf esetén minden élt oda-vissza irányított élnak tekintünk.) A bejárás során minden csúcsot elérünk egyszer (ez adja az elérési sorrendet), és minden csúcs bejárását befejezzük egyszer, mégpedig akkor, amikor nem érhető el belőle újabb eléretlen csúcs. Minden csúcshoz megjegyezzük azt is, hogy melyik élen értük el. Ez utóbbi élek alkotják a bejárás *fáját*, élei a bejáráshoz tartozó *faélek*. A fában ősből leszármazottba vezető él az *előreél*, a leszármazottból ősbé vezető a *visszaél*, a többi pedig a *keresztél*.

Def: A *szélességi bejárás* (BFS) inputja a $G = (V, E)$ gráf és egy v_0 csúcs. Az output egy v_0 -ból induló bejáráshoz tartozó ún. *szélességi fa* (a bejárás fája) és elérési sorrend. A bejárás az általános elv szerint történik azzal, hogy a következőnek elért csúcsot mindig a lehető legkorábban elért csúcsból kell elérnünk. (Ezért a BFS bejáráshoz tartozó elérési sorrend megegyezik a befejezési sorrenddel.)

Állítás: BFS bejárás után a G gráfban nincs előreél, ha G irányítatlan, akkor visszaél sincs. BFS megvalósítása FIFO sorral: a sorban az elért, de még bejáratlan csúcsok állnak, eredetileg egyedül v_0 . A sor elején található bejáratlan csúcs v_i eléretlen szomszédait a sor végére tesszük, ha nincs több ilyen csúcs, akkor v_i bejárt lesz, és kikerül a sorból.

Tétel: Ha V_i jelöli a v_0 csúcstól i távolságra levő csúcsok halmazát, akkor a BFS során minden V_i -beli csúcsot egy V_{i-1} -beli csúcsból érünk el. Ezért a szélességi fa a v_0 csúcsból minden más csúcsba a G gráf egylegrövidebb (legkevesebb élből álló) legrövidebb útját tartalmazza, azaz tetszőleges v csúcs G -beli távolsága v_0 -tól megegyezik a v_0 gyökerű szélességi fán mért távolsággal.

Tétel: A szélességi bejárás lépésszáma legfeljebb $\text{konst} \cdot n \cdot m$, ahol n a G csúcsainak, m pedig G éleinek száma.

Def: Adott $G = (V, E)$ (ir) gráf és egy $l : E \rightarrow \mathbb{R}$ élhosszfv. Egy G -beli (ir) út hossza az út éleinek összhossza. $\text{dist}(u, v)$ jelöli az (ir) uv utak közül a legrövidebb hosszát. Az l hosszfv *konzervatív*, ha nincs G -ben negatív összhosszú (ir) kör.

Def: Adott $G = (V, E)$ (ir) gráf, $u \in V$ és egy $l : E \rightarrow \mathbb{R}$ konzervatív élhosszfv. Tegyük fel, hogy $d(v) \geq \text{dist}(u, v)$ teljesül G minden v csúcsára. Az $e = vw$ él *menti javítás* azt jelenti, hogy a $d(w)$ értéket a $\min\{d(w), d(v) + l(vw)\}$ értékkel helyettesítjük. (Világos, hogy $d(w) \geq \text{dist}(u, w)$ az él menti javítás után is teljesülni fog.)

Dijkstra algoritmus

Input: $G = (V, E)$ (ir) gráf, $l : E \rightarrow \mathbb{R}_+$ nemneg hosszfv, $u_1 \in V$ gyökér.

Output: $\text{dist}(u_1, v)$ minden $v \in V$ -re. Működés: Kezdetben $U_1 := \{u_1\}$, $d(u_1) = 0$ és $v \neq u_1$ esetén $d(v) = \infty$. Az algoritmus i -dik fázisában végezzünk él menti javításokat minden U_i -ből kivezető $u_i v$ élen, majd válasszuk ki $V \setminus U_i$ -ből azt az v csúcsot, amire $d(v)$ minimális. Legyen ez a csúcs u_{i+1} . Jöhet az $(i + 1)$ -dik fázis. Az n -dik fázis után $\text{dist}(u, v) = d(v)$ teljesül minden $v \in V$ -re.

Tétel: A Dijkstra algoritmus helyesen működik. Minden fázis legfeljebb n javításból és egy minimumkiválasztásból áll (ami legfeljebb $\text{konst} \cdot n$ lépés), ezért a Dijkstra algoritmus lépésszáma legfeljebb $\text{konst} \cdot n^2$.

Megjegyzés: Ha minden v csúcsnál azt is nyilvántartjuk, melyik él menti javítás állította be a $d(v)$ értéket, akkor az így megjelölt élek a BFS-hez hasonló legrövidebb utak fáját alkotják.

Ford algoritmus Input: $G = (V, E)$ (ir) gráf, $l : E \rightarrow \mathbb{R}$ konzervatív hosszfv, $u \in V$ gyökérpont.

Output: $\text{dist}(u, v)$ minden $v \in V$ -re. Működés: Legyen $E = \{e_1, e_2, \dots, e_m\}$. Kezdetben $d(s) = 0$ és $v \neq s$ esetén $d(v) = \infty$ $i = 1$. (*) Végezzük el az $e_1, e_2, \dots, e_m$ menti javításokat. Ha $i < n - 1$, akkor $i := i + 1$ és GOTO (*), különben STOP: $\text{dist}(u, v) = d(v)$ minden v -re.

Floyd algoritmus Input: $G = (V, E)$ (ir) gráf, $l : E \rightarrow \mathbb{R}$ konzervatív. Output: $dist(u, v) \forall u, v \in V$. Működés: Legyen $V = \{v_1, v_2, \dots, v_n\}$ és $d^{(k)}(v_i, v_j)$ a legrövidebb olyan $v_i v_j$ út hossza, aminek belső pontjai csak $v_1, v_2, \dots, v_i$ lehetnek. Kezdetben $d^{(0)}(v_i, v_j) = l(v_i, v_j)$, ha $v_i v_j \in E$, különben $d^{(0)}(v_i, v_j) = \infty$. A k -dik fázisban $d^{(k)}(v_i, v_j) = \min\{d^{(k-1)}(v_i, v_j), d^{(k-1)}(v_i, v_k) + d^{(k-1)}(v_k, v_j)\}$ alapján a $d^{(k)}$ függvényt határozzuk meg. Ha $k = n$, akkor STOP: $dist(v_i, v_j) = d^{(n)}(v_i, v_j) \forall v_i, v_j$.

Def: Ha $G = (V, E)$ gráfon $w : E \rightarrow \mathbb{R}_+$ szélességfüggvény, akkor egy P út szélessége $w(P) := \min\{w(e) : e \in E(P)\}$ a legkeskenyebb P -beli él szélessége.

Tétel: Ha G irányítatlan és w egy szélességfüggvény, akkor a Kruskal algoritmust az élek csökkenő szélesség szerinti sorrendjében futtatva egy legszélesebb utak fáját kapjuk: bármely két csúcs között a fabeli út egy legszélesebb út G -ben.

Állítás: Ha G irányított, akkor egy v_0 gyökérből minden más csúcsba megkaphatunk egy legszélesebb utat a Dijkstra algoritmus alábbi módosításával.

Def: A legszélesebb utakat megtaláló Dijkstra algoritmus. Input: $G = (V, E)$ irányított gráf, $w : E \rightarrow \mathbb{R}_+$ szélességfüggvény és $u \in V(G)$ gyökér. Output: Minden v csúcsra egy legszélesebb uv út.

Működés: Kezdetben $U = \{v\}$.

1. Minden $v \in V(G) \setminus U$ pontra kiszámítjuk a legszélesebb olyan uv utat, aminek v kivételével minden csúcsa U -beli. Ehhez össze kell hasonlítani az eddig nyilvántartott uv út szélességét azzal az uv úttal, amit úgy kapunk, hogy a legutóbb U -hoz vett w csúcsba vezető legjobb uw utat kiegészítjük a wv éllel.

2. Azt a v csúcsot, amibe az imént kiszámított utak legszélesebbike vezet, U -hoz csatoljuk, és visszatérünk az 1. lépésre. Ha nincs ilyen v csúcs, akkor az algoritmus véget ért. Outputként minden csúcsra megadjuk, hogy melyik csúcsból értük el a legszélesebb úton.

Gyakorlatok

1. Törpöfalván kitört a járvány: ronda kórság fertőzött meg néhány törpöt. Szerencsére a betegségből minden törp egy nap alatt meggyógyul, és ezután egy napig immunissá válik, ám sajnos ezt követően újra fertőződhet. Kellemetlen, hogy a törpök még betegen sem adják fel azt a megrögzött szokásukat, hogy minden egyes nap minden barátjukat meglátogatják. Márpedig ha beteg és nem immunis törp találkozik, az utóbbi bizonyosan megfertőződik. Mutassuk meg, hogy ha Törpöfalván 100 törp él, akkor a járválynak a kitörését követő 101-dik napon már bizonyosan vége van.
2. Adjunk hatékony algoritmus, aminek a bemenete egy n csúcsú összefüggő irányítatlan gráf, a kimenet pedig egy olyan gráfcsúcs, amiből minden más csúcs lefeljebb $n/2$ élű úton elérhető.
3. Adott egy $n \times k$ méretű táblázat, amiben minden mezőben egy 0 vagy egy 1 áll. Találjunk a táblázat bal felső sarkától a jobb alsó sarokig egy mezőhatárok mentén jobbra és lefelé haladó olyan vonalat, amire az igaz, hogy a vonal alatti 1-esek és a vonal feletti 0-k számának összege a lehető legkisebb. Hogyan érdemes eljárni?
4. Tegyük fel, hogy a G irányítatlan gráf tetszőleges szélességi kereséssel kapott feszítőfája csillag. Mit lehet mondani G -ről?
5. Adott egy $G = (V, E)$ gráf, egy $l : E \rightarrow \mathbb{R}_+$ hosszfüggvény, valamint egy v gyökérpont. Egyetlen Dijkstra algoritmus lefuttatása segítségével találjuk meg G mindazon e éleit, amelyekre igaz az, hogy önmagában attól, hogy e hosszát eggyel csökkentjük egytlen csúcs v -től mért távolsága sem csökken.
6. Rajzoljunk gráfot, és keressük meg egy csúcsból kiindulva a BFS fáját. Megfelelő élhosszok megadásával gyakoroljuk a Dijkstra, Ford és Floyd algoritmusokat. Keressünk legszélesebb utakat irányított és irányítatlan gráfokban.