0.1 Általános gráfbejárás & BFS

A gráfbejárási algoritmus az inputgráf csúcsait és éleit fedezi fel. Minden csúcs az eléretlen → elért → befejezett állapotokat veszi fel. A bejárás akkor ér véget, amint minden csúcs befejezetté vált.

1. Van elért csúcs. Választunk egyet, mondjuk u-t.
(1a) Ha van olyan uv él, amire v eléretlen, akkor v elérté válik.
(1b) Ha nincs ilyen uv él, akkor u befejezetté válik.
2. Nincs elért csúcs.
(2a) Ha van eléretlen u csúcs, akkor u-t elértté tesszük.
(2b) Ha nincs eléretlen csúcs (azaz $\forall$ csúcs fejezett), akkor END.

Szélességi bejárás (BFS) szabálya:
Az 1. esetben mindeg a legkorábban elért u-t választjuk.
Input: $G = (V, E)$ (ir/ir.tatlan) gráf, ($v \in V$ gyökérpont[1]).
Output: (1) A csúcsok elérési és befejezési sorrendje. (2) Az élek osztályozása:
faél: Olyan él, ami mentén egy csúcs elértté vált.
uv **előreél:** nem faél, de u-ból v-be faélekből irányított út vezet.
uv **visszaél:** v-ből u-ba faélekből irányított út vezet.
keresztél: minden más él (u és v közt nincs leszármazott viszony).
(3) A **bejárás fája:** a faélek alkotta részgráf. (A bejárás fája valójában egy gyökereiből kifelé irányított erdő.)
Megf: Irányítatlan esetben az előreél és a visszaél ugyanazt jelenti.
Terminológia: Ha a bejárás fájában u-ból v-be irányított út vezet, akkor u a v őse és v az u leszármazottja. A faél és az előreél tehát őszből leszármazottba, a visszaél leszármazottból ősbe vezet.

0.2 A BFS tulajdonságai

Nézzük meg egy **irányított** gráf BFS bejárását is.
Állítás: Tfh $G = (V, E)$ BFS bejárása után a csúcsok elérési sorrendje $v_1, v_2, \ldots, v_n$. Ekkor az alábbiak teljesülnek.
(1) Ha $i < j$, akkor v_i-t hamarabb fejezük be, mint v_j-t, továbbá v_i gyerekei megelőzik v_j gyerekeit az elérési sorrendben.
(2) **Az elérési és befejezési sorrend (BFS esetén) megegyezik.**
(3) **Gréfél nem ugorhat át falét:** ha $k < i < j \leq l$ és v_iv_j faél, akkor v_kv_l nem lehet gráfél.
(4) **Nincs előreél.** (Irányítatlan eset: csak faél és keresztél van.)
(5) Ha a BFS-fában k-élű irányított út vezet u-ból v-be, akkor G-ben nincs k-nál kevesebb élű uv-út.
(6) **A BFS-fa egy legrövidebb utak fája:** a BFS-fa v_1 gyökeréből bármely v_i csúcsba vezető faút a G egy legkevesebb élű v_1v_i-útja.

0.3 Legrövidebb utak

Def: Adott G (ir) gráf és $l : E(G) \to \mathbb{R}$ hosszfüggvény esetén egy P út hossza a P éleinek összhossza: $l(P) = \sum_{e \in E(P)} l(e)$.

Az u és v csúcsok távolsága a legrövidebb uv-út hossza: $dist_l(u, v) := \min\{l(P) : P\ uv\text{-út}\}$ ($\nexists uv$-út$\Rightarrow dist_l(u, v) = \infty$.) Az l hosszfüggvénye nemnegatív, ha $l(e) \geq 0$ teljesül minden e élre. Az l hosszvüggvény konzervatív, ha G-ben $\nexists$ negatív összhosszú ir. kör.

[1] A gyökérben kezdetben elért állapotú, ezért kivétel az általános szabály alól.

Cél: Legrövidebb út keresése irányított/irányítatlan gráfban.

Megf: Ha $l(e) = 1$ a G minden e élére, akkor $l(P)$ a P élszáma. Ezért a BFS-fa minden gyökérből elérhető csúcsba tartalmaz egy legrövidebb utat a gyökérből elérhető csúcsba tartalmaz egy legrövidebb utat a gyökérből, azaz a szélességi bejárás tekinthető egy legrövidebb utat kereső algoritmusnak is.

Def: Adott G (ir) gráf, $l : E(G) \to \mathbb{R}$ hosszfüggvény és $r \in V(G)$. (r, l)-felső becslés olyan $f : V(G) \to \mathbb{R}$ függvény, ami felülről becsli minden csúcs r-től mért távolságát: $dist_l(r, v) \geq f(v) \forall v \in V(G)$.

Triviális (r, l)-felső becslés: $f(v) = \begin{cases} 0 & v = r \\ \infty & v \neq r \end{cases}$

Pontos (r, l)-felső becslés: $f(v) = dist_l(r, l)\ \forall v \in V(G)$.

0.4 Az elméleti javítás

Def: Tfh f egy (r, l)-felső becslés és $uv \in E(G)$. Az f uv-elméleti javítása az az f', amire

$$f'(z) = \begin{cases} f(z) & z \neq v \\ min\{f(v), f(u) + l(uv)\} & z = v \end{cases}$$

Megf: Tfh az $l : E(G) \to \mathbb{R}$ hosszfüggvény konzervatív és $f(r) = 0$.

Ekkor (1) Az $f(r, l)$-felső becslés élmenti javítása mindig (r, l)-felső becslést ad.

(2) $f(r, l)$-felső becslés (pontosan) $\iff$ (f-en $\nexists$ érdemi élmenti javítás).

Köv: Adott G, konzervatív l és $r \in V(G)$ esetén ha kiindulunk a triviális (r, l)-felső becslésből, és addig végzünk émj-kat, amíg lehet, akkor a végén megkapjuk minden csúcs r-től való távolságát.

Itt a jegyzet 17. oldaláról az utolsó kettő pont hiányzik, mivel nem tudom, hogy mennyire lényegesek.

Def: Tfh f egy (r, l)-felső becslés és $uv \in E(G)$. Az f uv-élmenti javítása az az f', amire

$$f'(z) = \begin{cases} f(z) & z \neq v \\ min\{f(v), f(u) + l(uv)\} & z = v \end{cases}$$

Megf: Tfh az $l : E(G) \to \mathbb{R}$ hosszfüggvény konzervatív és $f(r) = 0$. Ekkor (1) Az $f(f, l)$-felső becslés élmenti javítása mindig (r, l)-felső becslést ad. (2) $f(r, l)$-felső becslés (pontosan) $\Leftrightarrow$ (f-en $\nexists$ érdemi élmenti javítás).

Dijkstra-algoritmus: Input: $G = (V, E), l : E \to \mathbb{R}_+, r \in V$. Output: $dist_l(r, v) \forall v \in V$
Működés: $U_0 := \emptyset$, f_0 a triviális. (r, l)-felső becslés.

Az i-dik fázis:

1. Legyen $U_i := U_{i-1} \cup \{u_i\}$, ahol u_i olyan csúcs a $V \setminus U_{i-1}$ halmazból, amelyre $f_{i-1}(v)$ minimális.

2. $f_i : f_{i-1}$ élmenti javítása minden U_i-ből kivezető $u_i x$ élen. Output: $f_{|V|}$. Megjelöljük a végső $f_{|V|}(V)$ értékeket beállító éleket.

0.5 Dijkstra, egy példán

Dijkstra-algoritmus: Input: $G = (V, E), l : E \to \mathbb{R}_+, r \in V$. Output: $dist_l(r, v) \forall v \in V$
Működés: $U_0 := \emptyset$, f_0 a triviális. (r, l)-felső becslés.

Az i-dik fázis:

1. Legyen $U_i := U_{i-1} \cup \{u_i\}$, ahol u_i olyan csúcs a $V \setminus U_{i-1}$ halmazból, amelyre $f_{i-1}(v)$ minimális.

2. $f_i : f_{i-1}$ élmenti javítása minden U_i-ből kivezető $u_i x$ élen. Output: $f_{|V|}$. Megjelöljük a végső $f_{|V|}(V)$ értékeket beállító éleket.

Megf: Ha a v-be vezet megjelölt él, akkor vezet r-ből v-be megjelölt éleken út, és ennek hozza megegyezik $f_{|V|}(v)$-vel.

Köv: Ha a Dijsktra-algoritmus helyes, akkor az algoritmus végén a megjelölt élek egy legrövidebb utak fáját alkotják r gyökérrel.

0.6 Dijkstra helyessége

Megf: Tfh $u_1, u_2, \ldots, u_n$ a G csúcsainak sorrendje a Dijkstra-algoritmus végrehajtása után.
(1) Ekkor $f_{|V|}(u_i) \leq f_{|V|}(u_{i+1})$ teljesül $\forall 1 \leq i \leq n$.
(2) $f_{|V|}(u_1) \leq f_{|V|}(u_2) \leq \cdots \leq f_{|V|}(u_n)$
(3) A Dijsktra-algoritmus outputjaként kaptt $f_{|V|}$-n élmenti javítás nem tud változtatni.

Tétel: A Dijsktra-algoritmus helyesen működik, azaz G minden csúcsára igaz, hogy $dist(r, v) = f_{|V|}(v)$.

„Lépésszámanalízis”: Ha a G gráfnak n csúcsa és m éle van, akkor a Dijkstra-algoritmus n-szer keresi meg legfeljebb n szám minimumát, ami összességében legfeljebb $konst \cdot n^2$ lépést igényel. Ezen kívül legfeljebb m élmenti javítást véges, ami $konst' \cdot m$ lépés. Összességében tehát legfeljebb $konst'' \cdot (n^2 + m)$ lépésre van szükség, az algoritmus hatékony.