Gráfút hossza, gráfcsúcsok távolsága, nemnegatív és konzervatív hosszfüggvény, triviális és pontos (r,l)**-felső becslés, élmenti javítás. Dijkstra-algoritmus működése**, Ford-algoritmus helyessége és lépésszáma. Legrövidebb utak fájának létezése.

- **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.

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

- Adott $G = (V,E)$ irányított gráf és egy $l : E \to \mathbb{R}$ élhosszfv. Egy G-beli irányított út hossza az út éleinek összhossza, $dist_l(n, v)$ pedig az irányított uv-utak közül a legrövidebb hosszát jelöli.

  Az l hosszfv konzervatív ha nincs G-ben negatív összhosszúságú irányított kör.

  Adott $G = (V,E)$ irányított gráf $r \in v$ és egy $l : E \to R$ élhosszfv. Az $f : v \to R$ függvényt (r,l)-felső becslésnek nevezzük, ha $f(r) = 0$ és $f(v) \leq dist_l(r, v)$ teljesül G minden v csúcsára. Az $e = uv$ élmenti javítás esetén a $f(v)$ értéket a $min\{f(v), f(u) + l(uv)\}$ értékkel helyettesíthetjük.

  (1) Ha l konzervatív akkor tetsz. (r,l)-f. b. élmenti javítása (r,l)-fb-t ad.

  (2) Ha az $f(r,l)$ felső becsléshez nincs érdemi élmenti javítás, akkor $f(v) = dist_l(r, v)\ \forall v \in V$.

- **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.

  Biz: Azt kell megmutatni, hogy van olyan rv-út, aminek a hossza legfeljebb $f(u) + l(uv)$. Ha egy legrövidebb ru-utat kiegészítünk az uv éllel, akkor olyan rv-élsorozatot kapunk, aminek az összhossza $dist_l(r, u) + l(uv) \leq f(u) + l(uv)$. „Könnyen" látható, hogy az élhosszfüggvény konzervativitása miatt ha van x összhosszúságú rv-élsorozat, akkor van legfeljebb x összhosszúságú rv-út is. Ezek szerint van legfeljebb $f(u) + l(u, v)$ hosszúságú uv-út is, azaz az érdemi élmenti javítás után szintén (r,l)-felső becslést kapunk. □

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

  Biz: $\Rightarrow$: Ha f pontos, akkor biztosan nincs rajta érdemi élmenti javítás: ha volna, akkor egy felső becslés a pontos érték alá csökkenne, így az élmenti javítás nem (r,l)-felső becslést eredményezne. $\Leftarrow$: Legyen $v \in V(G)$ tetsz, és legyen P egy legrövidebbb rv-út. A P egyik éle mentén sincs érdemi élmenti javítás, ezért P minden u csúcsára pontos a felső becslés: $f(u) = dist_l(r, u)$. Ez igaz az út utolsó csúcsára, a tetszőlegesen választott v-re is. □

- Dijkstra algoritmus működése:

  - **Input**: $G = (V,E)$ irányított gráf, $l : E \to \mathbb{R}^+$ nemnegatív hosszfüggvény, $r \in V$ gyökér
  - **Output**: $dist_l(r, v)$ minden $v \in V$-re.
  - **Működés**: Kezdetben $U_0 = \emptyset, f(r) = 0$ és $f(v) = \infty$, ha $v \neq r$.
    Az algoritmus i-dik fázisában $(i = 1, 2, ..., |v|)$ a következő történik.

    1. Legyen u_i az v csúcs a $v \backslash u_{i-1}$ halmazból, amelyre $f(r)$ minimális és legyen $u_{i-1} \cup u_i$.

2. Végezzünk élmenti javításokat minden u_i-ből kivezető $u_i x$ élen.

Az output a $|v|$-dik fázik utáni f függvény. Szokás megjelölni a végső $f(v)$ értékeket beállító éleket. Ha az output az $f(r,l)$-felső becslés, akkor

(1) $f(u_i) \leq f(u_{i+1})$ $\forall 1 \leq u$-re.

(2) $f(u_i) \leq f(u_2) \leq \cdots \leq (u_n)$

(3) élmentijavítás nem változhat f-n.

- A Dijkstra-algoritmus helyesen működik, azaz $dist_l(r,v) = f(v) \forall v \in V$ teljesül. Az algoritmus során megjelölt élek egy legrövidebb utak fáját alkotják G-ben: az r gyökérből minden r-ből elérhető csúcshoz vezet olyan legrövidebb út is ami csak megjelölt éleket tartalmaz.
- A Dijkstra-algoritmus lépésszáma legfeljebb. $konst \cdot (n^2 + m)$, ahol $n = |v|$ $m = |E|$.

- **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.

  Biz: $f_{|V|}(r) = 0$, és a megjelölt élek mentén haladva az $f_{|V|}$ érték az élhosszal növekszik. $\square$

- **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 $\forall 1 \leq i \leq n$.

  Biz: Az i-dik fázisban $f_i(u_i) \leq f_i(u_{i+1})$ teljesült az u_i választása miatt. Ezek után $f_i(u_i)$ már nem változott: $f_{|V|}(u_i) = f_i(u_i)$. Ugyan $f_i(u_{i+1})$ még csökkenhetett, de csak az $u_i u_{i+1}$ él mentén történt javítás miatt, hiszen az $(i+1)$-dik fázisban u_{i+1} bekerült az U_i halmazba, és a hozzá tartozó (r,l)-fb már nem csökken tovább. Ekkor $f_{i+1}(u_{i+1}) = min\{f_i(u_{i+1}), f_i(u_i) + l(u_i u_{i+1})\} \geq f_i(u_i)$, mivel $l(u_i u_{i+1}) > 0$. Ezért $f_{|V|}(u_i) = f_i(u_i) \leq f_{i+1}(u_{i+1}) = f_{|V|}(u_{i+1})$ $\square$

  (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.

  Biz: Tegyük fel, hogy $u_i u_j \in E(G)$ a G egy tetszőleges éle. Ha $i > j$, akkor (2) miatt $f_{|V|}(u_i) \geq f_{|V|}(u_j)$, ezért az $u_i u_j$ mentén történő javítás nem tudja $f_{|V|}(u_j)$-t csökkenteni, hisz $l(u_i u_j)$ pozitív. Ha pedig $i < j$, akkor az i-dik fázisban megrörtént az $u_i u_j$ mentén történő javítás, és ezt követően $f(u_i)$ nem váltorott, azaz $f_{|V|}(u_i) = f_i(u_i)$. A másik (r,l)-felső becslés pedig csak tovább csökkenhetett a késpbbi émj-ok során $f_{|V|}(u_j) \leq f_i(u_j)$. Ezért az $u_i u_j$ él mentén sem az i-dik fázisban, sem később nincs érdemi javítás. $\square$

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

  Biz: A Dijsktra-algoritmus az f_0 triviális (r,l)-felső becslésből indul ki, és élmenti javításokat alkalmaz. Így minden f_i (speciálisan $f_{|V|}$ is) (r,l)-felső becslés lesz. A fenti (3)-as megfigyelés miatt $f_{|V|}$-n nem végezhető érdemi élmenti javítás. Ezért egy korábbi (2)-es megfigyelés miatt $f_{|V|}$ pontos (r,l)-felső becslés, azaz $f_{|V|}(v) = dist_l(r,v) \forall v \in V(G)$. $\square$

- Könnyű olyan példát találni, ahol a Dijkstra-algoritmus konzervatív hosszüggvény esetén hibás eredményt ad. Azonban konzervatív hosszüffvény esetén is igaz, hogy

  - (r,l)-fb élmenti javítása (r,l)-fb-t eredményez, ill.
  - ha egy (r,l)-fb-ben nem végezhető erdemi élmenti javítás, akkor pontos.

  konzervatív hosszfüggvény esetén is hasonló startégiát követünk: Élmenti javításokat végzünk a triviális (r,l)-fb-en, míg van érdemi javítás.

  Ford-algoritmus:
  Input: $G = (V,E)$ irányított, $l : E \to \mathbb{R}$ konzervatív hosszfüggvény, $r \in V$ gyökérpont.
  Output: $dist_l(r,l)$ minden $v \in V$
  Működés: Legyen $E = \{e_1, e_2, \ldots, e_m\}$. Kezdetben legy $f(r) = 0$ és $v \neq r$ esetén $f(v) = \infty$, Az i-dik fázis $i = 1, 2, \ldots, n-1$ esetén abból áll, hogy elvégezzük az $e_1, e_2, \ldots, e_m$ élek menti javításokat. A végén az OUTPUT: $dist_l(r,v) = f(v)$ minden v-re. ($dist_l(r,v) = f_{n-1}(v) \forall v \in V$)

Állítás: Ha l konzervatív, akkor $dist_l(v)$ $v \in V$-re.

Biz: $f_1(v) = dist_l(r, v)$ ha $\exists \leq 1$-élű legrövidebb rv-út. $f_2(v) = dist_l(r, v)$ ha $\exists \leq 2$-élű legrövidebb rv-út. $\ldots$ $f_{n-1}(v) = dist_l(r, v)$ ha $\exists \leq (n-1)$-élű legrövidebb rv-út. Tehát $f_{n-1}(v) = dist_l(r, v) \forall v \in V$. $\square$

Megf: Ha $f_i = f_{i-1}$, akkor a Ford-algoritmust az i-dik fázis után be lehet fejezni, hisz nincs érdemi élmenti javítás, így $f_{n-1} = f_i$.

Megj: Az $f_{n-1}(v)$-t beállító élek legrövidebb utak fáját alkozják.

Biz: A Dijkstra esethez hasonló. Tetszőleges v csúcsból visszafelé követve a végső értékeket beállító éleket $f_{n-1}(v)$ hosszúságú rv-utat találunk. $\square$

"Lépésszámanalízis": Ha a $|V(G)| = n$ és $|E(G)| = m$, akkor minden fázisban $\leq m$ élmenti javítás, ami $konst \cdot m$ lépés. Ez összesen $\leq konst \cdot (n-1) \cdot m \leq konst \cdot n^3$ lépés, az algoritmus hatékony.