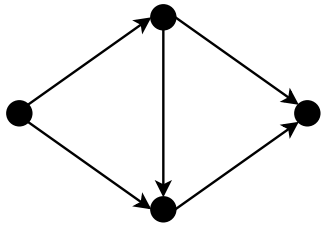
Algel IV. gyakorlat

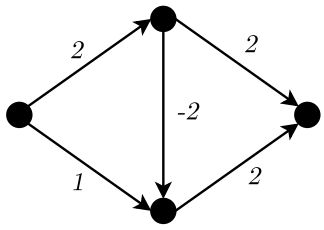
Még mindig legrövidebb utak

2009. március 3/5.

1. **Határozzuk meg a következő gráfban az élsúlyokat úgy, hogy a Dijkstra algoritmus rossz eredményt adjon!**



Attól, hogy rakunk bele negatív élsúlyt, még akár működhetne! Úgy kell negatív élsúlyt belerakni, hogy romoljon el, pl:



Az alsó csúcsra az alogritmus 1-et mond, pedig a legrövidebb út oda 0 lenne.

3. **Éllistával adott a súlyozott élű $G(V, E)$ gráf. Tegyük fel, hogy az élek súlyai az $1, 2, 3$ számok közül valók. Javasoljunk $O(n + e)$ költségű algoritmust az $s \in V$ pontból az összes további $v \in V$ pontokba vivő legrövidebb utak hosszának meghatározására!**
A 2 hosszú éleket helyettesítsük 2, a 3 hosszúakat 3 élellel. Az utak hossza így az élszám lesz (minden út annyi élből áll így, amilyen hosszú az eredeti gráfban lenne), így a szélességi bejárás használható legrövidebb út keresésére. A lépésszám $O(3n + 3e) = O(n + e)$.

5. **Egy G gráfban pontosan egy él súlya negatív, és nincs a gráfban negatív összsúlyú irányított kör. Adjunk $O(n^2)$ lépésszámú algoritmust az $s \in V(G)$ pontból az összes többi pontba vezető legrövidebb utak meghatározására!**
Egy legrövidebb út kétféle lehet: vagy használja a negatív élet, vagy nem. Először futtassunk egy Dijsktrát (a szépség kedvéért hagyjuk ki a negatív élet), az i csúcsba vezető legrövidebb út hosszát jelölje d_i^+. Jelölje u azt a csúcsot, amiből a negatív él indul. Világos, hogy d_u^+ helyes, hiszen a negatív élen nem mehetünk át az ő eléréséhez az eredeti gráfban. Belőle is indíthatunk egy Dijsktrát, hiszen ekkor először a negatív él másik végpontját veszi be először a KÉSZ halmazba, ami a definíciónak megfelel. Ezeket a legrövidebb értékeket jelölje d_i^-. A legrövidebb utak tehát $\min(d_i^+, d_u^+ + d_i^-)$ képlettel számolhatók. Lépésszám egy keresés és két Dijsktra, azaz $O(n + n^2 + n^2) = O(n^2)$.

6. **[ZH: 2007. április 27.] Kutyasétáltatáskor egy parkban egy gazda rögzített, egyenes szakaszokból álló útvonalon halad, aminek töréspontjai $t_1, \ldots, t_n$, a bejáratot jelölje t_0, a kijáratot t_{n+1}. A kutyája szabadon szaladgál, de a t_i pontokban találkozik a gazdájával. A t_i és t_{i+1} pontokban való találkozás között a kutya szeretne egy fát is meglátogatni (minden $i = 0, 1, \ldots, n$ esetén legfeljebb egyet-egyet). Legyenek adottak az $s(t_i, t_{i+1})$ távolságok $(0 \leq i \leq n)$, valamint minden fának az összes t_i ponttól vett távolsága. Tegyük fel, hogy két találkozás között a kutya legfeljebb kétszer akkora távolságot tud megtenni, mint a gazda. Adjon algoritmust, ami**

segít a kutyának eldönteni, hogy mikor melyik fát látogassa meg ha a kutya célja, hogy minél több fánál járjon. Az algoritmus lépésszáma legyen $O(n^2f + nf^2)$, ahol f a parkban levő fák számát jelöli.

Ez csak annyiban trükkös, hogy nem legrövidebb út, hanem párosítás. Ugyanis minden gazdaponthoz párosítani akarunk egy fát. Egyik pontosztály tehát a gazda pontjai, másik a fák, akkor megy él, ha adott pontból elérhető a fa úgy, hogy utána a kutya vissza tud érni. Itt kell max. párosítás, ami mehet magyar módszer, ami $O(|V||E|) = O((n+f)(nf)) = O(n^2f+nf^2)$, a felépítés pedig megy $O(nf)$-ben, ami nyilván belefér.