

Algel III. gyakorlat

Párosan utazunk

2012. február 22.

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 élel. 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 $O(3n + 3e) = O(n + e)$.

6. Legfeljebb hány komponensből állhat egy irányított gráf szélességi bejárása során keletkező erdő?

Vegyünk egy n hosszú irányított utat! Ha a szélességi bejárást az utolsó pontból indítjuk, akkor elakad, utána az utolsó előttiből, ott is elakad stb., így pont n komponens fog csinálni. Ennél többet nem is lehet n pontból.

7. [ZH: 2009. április 24.] 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 a kereszteződés. (Az hogy nehéz, a kereszteződés tulajdonsága, nem azon múlik, merről érkezik oda és és merre akar rajta áthaladni.) Írjon le egy 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. Az algoritmus lépésszáma állítás megadás esetén legyen $O(n + e)$, ahol n a csúcsok és e az élek száma.

A gráfból kihagyjuk a nehezek közti éleket. Szélességi bejárással megkeressük, hogy mely csúcsok vannak az otthonnal egy komponensben, ezek lesznek elérhetők. Ez jó, mert pontosan azokat az éleket hagytuk el, amiken nem mehetünk át, minden más benne maradt (ezt lehet prezízebben is). Lépésszám: éltörlések $O(e)$, bejárás $O(n + e)$, ami összesen $O(n + e)$.

8. Legyen $G = (V, E)$ mátrixszal adott n pontú, súlyozott élű irányított gráf! Tegyük fel, hogy G nem tartalmaz negatív összhosszúságú irányított kört, továbbá azt, hogy a G -beli egyszerű irányított utak legfeljebb 25 élből állnak. Javasoljunk $O(n^2)$ költségű módszert az $1 \in V$ pontból az összes további $v \in V$ pontokba vivő legrövidebb utak hosszának a meghatározására!

A Bellmann-Ford algoritmus során a táblázat i -edik sorában a kezdőpontból a legfeljebb i élszámú legrövidebb utak hosszai szerepelnek (ezt nem kell külön bizonyítani!). Mivel tudjuk, hogy G -ben minden út legfeljebb 25 élből áll, a táblázatot elég a 25. sorig kitölteni. A lépésszám így $O(25n^2) = O(n^2)$ lesz.

9. Adott állítással egy n pontú, e élű G összefüggő irányítatlan gráf. Adjunk $O(e)$ költségű algoritmust olyan $X \subset V(G)$ központi ponthalmaz keresésére, melyre $|X| \leq n/2$ teljesül! Az $X \subseteq V(G)$ egy központi ponthalmaz, ha G minden pontja vagy X -beli, vagy egyetlen éllel elérhető valamelyik X -beli pontból.

Az összefüggőség és irányítatlanság miatt a szélességi bejárás egy szélességi feszítőfát fog adni. A páratlan szinteken lévő pontok száma legyen x , értelemszerűen a párosoké ekkor $n - x$. Ha $x < n - x$, akkor válasszuk központi halmaznak a páratlan szinten lévő pontokat, egyébként a párosakat. Könnyen látszik, hogy minden pontra igaz, hogy vagy benne van a kiválasztott halmazban, vagy elérhető belőle egy lépésben (elég csak a szomszéd szintre menni). A kiválasztott halmaz a skatulya-elv miatt biztos legfeljebb $n/2$ méretű lesz. A lépésszám $O(n + e)$, ami az összefüggőség miatt $O(e)$.

10. [ZH: 2007. április 27.] Kutyasétáltatáskor egy parkban egy gazda rögzített, egyenes szakaszokból álló útvonalon halad, aminek töréspontjai $t_1, \dots, t_n$, 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 $i = 0, 1, \dots, n$ 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 érní. Itt kell max. párosítás, ami mehet magyar módszerrel, ami $O(|V||E|) = O((n+f)(nf)) = O(n^2f + nf^2)$, a felépítés pedig megy $O(nf)$ -ben, ami nyilván belefér (a távolságszámítás egy pont és fa és pont között konstans idő).