Dijkstra-algoritmus

Algoritmuselmélet

6. gyakorlat

2024.

1. Egy irányított gráf csúcshalmaza $\{a, b, c, d, e, f\}$, az élek és súlyaik pedig az alábbiak: $s(a, b) = 5$, $s(a, e) = 6$, $s(b, c) = 4$, $s(b, d) = 6$, $s(c, a) = 3$, $s(c, d) = 1$, $s(d, e) = 2$, $s(e, c) = 2$, $s(e, f) = 1$, $s(f, b) = 3$, $s(f, c) = 1$, $s(f, d) = 1$. Dijkstra módszerével határozzuk meg a-ból az összes többi csúcsba vezető legrövidebb út hosszát.

2. Adjuk meg az összes olyan minimális élszámú irányított gráfot (élsúlyokkal együtt), amelyekre az alábbi táblázat a Dijkstra-algoritmusban szereplő D tömb változásait mutathatja.

v_1	v_2	v_3	v_4	v_5	v_6
0	∞	∞	∞	∞	∞
0	2	6	∞	∞	7
0	2	5	9	∞	6
0	2	5	6	9	6
0	2	5	6	8	6
0	2	5	6	7	6

3. Az alábbi táblázat a Dijkstra-algoritmus lefutását mutatja egy G irányítatlan gráfon.

   (a) Határozzuk meg, hogy milyen sorrendben kerültek be az egyes csúcsok a KÉSZ halmazba.

   (b) Határozzuk meg az ac él hosszát.

a	b	c	d	e
∞	∞	∞	0	∞
42	24	7	0	∞
33	16	7	0	77
24	16	7	0	18
22	16	7	0	18

4. Mátrixával adott egy város úthálózatának élsúlyozott, irányított gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak, az élek súlya pedig azt mutatja, hogy mennyi az átlagos idő, ami az út megtételéhez autóval szükséges. Útfelújítások miatt a következő héten le fogják zárni a város két csomópontját, a-t és b-t (ezeken nem lehet autóval áthaladni). Adott a gráfban két kijelölt csúcs, S és T és azt szeretnénk eldönteni, hogy az a és b csomópontok lezárása miatt növekedni fog-e az S-ből T-be eljutás ideje és ha igen, akkor mennyivel. (Tételezzük fel, hogy a közvetlen utakhoz rendelt átlagos idők nem változnak a lezárások következtében.) Melyik tanult algoritmust lehet alkalmazni, hogyan és miért, ha $O(n^2)$ lépésben meg akarjuk oldani ezt a feladatot, ahol n a csomópontok számát jelöli?

5. A mátrixával adott n-csúcsú, irányított G gráf élei között van egy negatív súlyú él, a többi él súlya pozitív. A gráfban nincs negatív súlyú kör. Adjunk $O(n^2)$ lépésszámú algoritmust az $s \in V(G)$ pontból az összes többi pontba vezető legrövidebb utak meghatározására.

6. A $G = (V, E)$ irányított gráfban a csúcsok egy nemüres F részhalmaza fontos. A gráf minden éléhez tartozik egy pozitív élsúly. Az $u \in F$ fontos csúcs távolsága a $v \in F$ fontos csúcstól a legrövidebb olyan u-ból v-be menő út hossza, aminek nincs u-tól és v-től különböző fontos csúcsa. Legyen a gráf a mátrixával adott, és minden csúcsra adott az is, hogy fontos csúcs-e. Adjunk algoritmust, ami $O\big(|V|^2|F|\big)$ lépésben meghatározza az összes fontos csúcspár közötti távolságot.

7. Mátrixával adott egy város úthálózatának összefüggő, élsúlyozott, irányított gráfja: a csúcsok a csomópontok, az élek a csomópontok közötti közvetlen utak, az élek súlya pedig azt mutatja, hogy mennyi idő alatt tud az adott szakaszon egy biciklis futár végigmenni. Egy, az f csúcsban tartózkodó biciklis futár azt a feladatot kapja, hogy a nála levő két csomagot a lehető leggyorsabban kézbesítse ki a város b és c csomópontjaiba (az mindegy, hogy milyen sorrendben kézbesít). Melyik tanult algoritmust lehet alkalmazni és hogyan, hogy $O(n^2)$ lépésben meghatározzuk, hogy milyen sorrendben kell a futárnak a csomagokat leadnia és mennyi a legrövidebb idő, ami alatt teljesíteni tudja a feladatát, ahol n a csomópontok számát jelöli?

8. Egy város úthálózata egy n-csúcsú irányítatlan gráffal adott, a közlekedési csomópontok a gráf csúcsai. Adott továbbá a szomszédos csomópontok közötti távolság. A városban $J < n$ buszjárat van. A járatok végállomásai és a megállói is csomópontokban vannak, egy járat minden megállója különböző. Adott minden járatra, hogy mely csomópontokban vannak az egymás utáni megállók. Két egymás utáni megálló nem biztos, hogy szomszédos pontban van – a busz nem áll meg minden utcasarkon. Viszont az egy csomópontban levő megállók egy helyen vannak, nem kell köztük gyalogolni. Amikor a város egyik pontjából a másikba akarunk eljutni, olyan útvonalat választunk, hogy utunk során összesen a lehető legkevesebbet kelljen gyalogolni (közben tetszőlegesen sokat buszozhatunk, az átszállások száma nem korlátozott). Adjunk $O(n^3)$ lépésszámú algoritmust, ami meghatároz két olyan csomópontot, amelyek között a feltételeknek megfelelő útvonalon a legtöbbet kell gyalogolni.

9. Űrhajónkkal egy véges, egydimenziós univerzumban ragadtunk, amelynek pontjait 0 és r közötti egész koordinátákkal azonosítjuk. A közlekedést a $0 \leq x_1, x_2, \ldots, x_n \leq r$ koordinátákon elhelyezett tükörteleportok segítik: egy-egy tükörteleport aktiválásakor űrhajónk pozíciója tükröződik a teleport pozíciójára, azaz például az y koordinátáról a $2x_i - y$ koordinátára jutunk (vagy megsemmisülünk, ha ezzel kilépnénk az univerzumból). Az s koordinátáról indulva szeretnénk a t koordinátán található – menekülést jelentő – féregjárathoz jutni. Adjon $O(r^2)$ lépésszámú algoritmust, amely meghatározza, hogy ehhez legkevesebb hányszor kell teleportálnunk.

10. Vidéken autózunk, ahol benzinkút csak bizonyos falvakban van. Az A falubeli benzinkúttól indulunk és a B faluba akarunk elérni (ahol szintén van benzinkút). A falvak közötti utakat egy n-csúcsú, m-élű, összefüggő, irányítatlan gráf írja le, melynek csúcsai a falvak, az élek pedig a falvak közötti utakat jelentik, egy él súlya a két falut összekötő útszakasz hossza. A gráf az éllistájával adott, és ezen kívül adott még az a k falu, amelyben van benzinkút. Adjunk $O(kn^2)$ lépésszámú algoritmust, amely meghatározza az A-ból B-be vivő legrövidebb olyan útvonalat, melynek során soha nem kell 600 kilométernél többet autóznunk két benzinkút között.

11. A húsvéti nyúl belefáradt, hogy mindenki ajándékot vár tőle. Ezentúl úgy jár el, hogy az első helyen, ahova megy, nem ad ajándékot, a második helyen ad, a következő helyen megint nem ad, és így tovább. Adott egy $G = (V, E)$ egyszerű irányított gráf, ami azt mutatja, hogy az x csúcsnak megfelelő helyről a nyúl következő lépése mely y csúcsokba vihet, az él súlya jelzi az átjutáshoz szükséges időt. Tegyük fel, hogy mátrixával adott a gráf, tudjuk, hogy a nyúl az $f \in V$ fészkéből indul, a mi helyzetünket az $m \in V$ csúcs jelzi. Adjunk $O(|V|^3)$ idejű algoritmust, amellyel meghatározhatjuk, hogy mi az a legkorábbi időpont, amikor a nyúl ajándékozó kedvvel érhet hozzánk. (A nyúl útja során egy csúcsot többször is meglátogathat és nem kell minden csúcsba eljutnia.)