

Dijkstra-algoritmus

ALGORITMUSELMÉLET

7. gyakorlat

2025.

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 $G = (V, E)$ 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 csúctól 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 $O(|V|^2|F|)$ lépésben meghatározza az összes fontos csúcspár közötti távolságot.

6. 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?
7. 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.
8. Űrhajónkkal egy véges, egydimenziós univerzumban ragadtunk, amelynek pontjait 0 és r közötti egész koordinátákkal azonosítjuk. A közlekedést a $0 \leq x_1, x_2, \dots, x_n \leq r$ koordinátákon elhelyezett tükröteleportok segítik: egy-egy tükröteleport 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 – féregjáráshoz jutni. Adjunk $O(r^2)$ lépésszámú algoritmust, amely meghatározza, hogy ehhez legkevesebb hányszor kell teleportálnunk.
9. Egy középkori királyság úthálózata egy n -csúcsú irányítatlan gráffal adott (a csúcsok a városok, az élek a köztük vezető utak). Az A városból szeretnénk a B városba árut vinni, de bizonyos városok csak akkor engednek át minket a terményünkkel, ha vámot fizetünk nekik (az A és B városban nem kell vámot fizetni). A vám összege fix, nem függ az áru mennyiségétől, de a vám városonként más és más lehet. Adjunk algoritmust, ami a városonkénti vámok és a gráf szomszédossági mátrixának ismeretében $O(n^2)$ lépésben talál olyan útvonalat, amin a legkevesebb sarcot szedik be tőlünk.
10. Éllistájával adott egy irányított gráf, mely egy városok közötti úthálózatot ír le: a csúcsok a városok, az élek a közvetlen utak és az élek súlya a városok közötti közvetlen útszakaszok hosszát adja meg. Egyes városokban nevezetességek is vannak, ezek száma minden városra meg van adva egy tömbben (a nevezetességek száma mindegyik városnál egy nemnegatív egész szám). Szeretnénk eljutni az A városból a B városba, és az elsődleges szempont, hogy az út a lehető legrövidebb legyen. De ha esetleg több egyforma hosszú legrövidebb út is van, akkor ezek közül azt akarjuk kiválasztani, ami során az érintett városokban a lehető legtöbb nevezetességet tudjuk megnézni. Adjunk $O(n^2)$ lépésszámú algoritmust egy ilyen út megkeresésére.
11. 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.
12. 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 $f \in V$ fészkeből 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.)