Gráfok: motiváció és alapfogalmak

Csima Judit
BME SZIT
csima@cs.bme.hu

2019. október 30.

Gráfok: motiváció

A félév hátrelevő részében gráfokkal, gráfalgoritmusokkal fogunk foglalkozni. Amellett, hogy megismerkedünk az alapfogalmakkal és alapvető algoritmusokkal, ennek a résznek a fő célja az, hogy a gráfokat, mint modellezési eszközt meg tanulják használni, valós életbeli feladatokat át tudjanak gráfokra fogalmazni, ahol szerencsés esetben már egy ismert algoritmus (amik közül néhányat tanulni is fogunk) már használható.

Valós példák, ahol gráfalgoritmusok hasznosak

1. Adottak szerverek, melyek kábelekkel vannak összekötve. Szeretnénk tudni, hogy az így kapott hálózat összefüggő-e, azaz igaz-e, hogy bámely szervertől bármely más szerverig van összeköttetés (nem feltételnül közvetlen). Hasonló kérdés, hogy egy összefüggő hálózatban néhány kábelt átvágva összefüggő marad-e az új hálózat is vagy sem.

   Ebben a helyzetben a hálózatot egy irányítatlan gráffal tudjuk leírni, ahol a gráf csúcsai a szerverek, a köztük levő kábelek pedig irányítatlan élek, a fenti kérdésekre pedig egy szélességi vagy mélységi bejárás tud választ adni. (Ezeket az algoritmusokat majd tanulni fogjuk.)

2. Adottak munkák, amiket el kell végezni és bizonyos feladat-párokra adott, hogy a második feladatot csak az első után lehetséges elvégezni. Cél az, hogy olyan sorrendbe rakjuk (ütemezzük) a munkákat, ahol mindegyik munka akkor kerüljön sorra, amikor már minden előfeltétele elkészült (vagy ismerjük fel, ha ilyen sorrend nem lehetséges). Ez a helyzet számos valós alkalmazásnál előállhat:

   - Lakásfelújítás vagy építés során az egyes részfeladatok a munkák, pl. kömüves lebontja az egyik falat, befalazza az egyik ajtót, a festő kifesti a szobát, stb.), előfeltétel lehet pl. az, hogy előbb kell megépíteni a falat, mint ahogy a villanyszerelő jön a vezetékeket behúzni, stb.)
   - Egy bonyolult gyártási folyamat sok kisebb részfeladatból áll, melyek között több megkötés is fennállhat.
   - Adatbáziskezelés során sok rövid program, úgynevezett tranzakció fut egyszerre, részben ugyanazokon az adatokon. Ilyen helyzet például egy banki rendszer, ahol a tranzakciók az egyes átutalások (az A számláról egy adott összeg átrakása egy

B számlára). A tranzakciókra alapvető megkötés, hogy egy tranzakció vagy minden utasítása végrehajtódik vagy egy sem, az előbbi példával élve vagy levonódik az összeg az A számláról és átkerül a B-re vagy nem történik semmi, az nem elfogadható, hogy csak az A számláról kerül le a pénz (de B-n nem jelenik meg), ahogy az sem lehetséges, hogy a B számlán úgy jelenjen meg az összeg, hogy A-ról nem történt levonás. Ahhoz, hogy ezt biztosítani lehessen a tranzakciók (ideiglenes) zárakat helyezhetnek el azokon a számlákon, amikkel dolgozni akarnak és csak akkor kezdik el az érdemi munkát, amikor mindent zárolni tudtak, amivel dolgozni akarnak. Amíg egy tranzakció zárat tart egy számlán, addig más tranzakciók nem férnek hozzá ezehez a számlákhoz. (Ez a legegyszerűbb modell, ennél bonyolultabb modellek is vannak.) Ez a zárolási stratégia azt eredményezi, hogy bizonyos tranzakcióknak várniuk kell más tranzakciókra, hogy azok elengedjenek zárakat. Ezt a helyzetet is felfoghatjuk úgy, hogy a munkák a tranzakciók és ha egy tranzakcióra vár egy másik, akkor az első tranzakciónak be kell fejeződnia, mielőtt a második elkezdhet dolgozni.

Ezekben a helyzetekben a munkák alkotják a gráf csúcsait, közöttük irányított élek mennek, az előfeltételeknek megfelelően (él van egy munkából egy másikba, ha az elsőt be kell fejezni a második előtt), az ütemezés megkeresésére (vagy annak belátására, hogy nincs jó ütemezés) a mélységi bejárást tudjuk majd alkalmazni.

3. Útvonaltervezés: Adott egy térkép (városokkal és köztük futó utakkal vagy egy város térképe, ahol adottak a csomópontok és a köztük futó közvetlen utcák (amik lehetnek egyirányúak is esetleg). Ismert továbbá, hogy az egyes közvetlen utakat (városok vagy csomópontok között) mennyi idő alatt lehet megtenni autóval. Feladat lehet az, hogy adott A pontból megkeressük a legrövidebb ideig tartó utat egy másik adott B pontba vagy az, hogy egy adott A pontból megkeressük a leggyorsabban elérhető valamilyen tulajdonságú másik csúcsot (pl. adott városból a legközelebbi benzinkutat).

   Itt a gráf csúcsai a városok vagy a várostérkép esetén a csomópontok, az élek pedig az ezek között futó közvetlen összeköttetéseknek felelnek meg (egyirányú utak esetén a gráf élei is irányítottak). Ebben a helyzetben az éleknek címkéje, úgy nevezett súlya is van, ahol a súly az adott szakasz megtételéhez szükséges idő, a legrövidebb út keresésére pedig számos algoritmust fogunk tanulni.

4. Bicikliút építése: adott egy város térképe, ahol minden útszakaszra, ami két csomópontot köt össze ismert, hogy mennyi pénzbe kerülne ide bicikliutat építeni. Célunk az, hogy kitaláljuk, hogy pontosan hol épüljenek bicikliutak, ha azt szeretnénk elérni, hogy a város bármely pontjáról bármely másik pontjára el tudjunk jutni bicikliúton és az építkezés összköltsége a lehető legkisebb legyen.

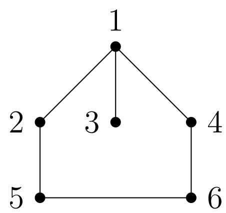
   Ezt a helyzetet egy irányítatlan, élsúlyozott gráffal tudjuk leírni, ahol csúcsok a csomópontok, élek a közvetlen utak ezek között, az él súlya pedig az adott szakasz építési költsége. A feladatot, amit meg kell oldanunk minimális feszítőfa keresésnek fogjuk hívni és tanulni fogunk erre is egy algoritmust.

Gráfokkal kapcsolatos alapfogalmak

Irányítatlan gráf: a G irányítatlan gráfot a V csúcshalmaz és E élhalmaz adja meg. A V csúcshalmaz az $1, 2, 3\ldots, n$ címkékkel megjelölt csúcsokból áll (bár néha a csúcsokat inkább betűkkel fogjuk jelölni), az élek pedig csúcspárokat kötnek össze (vagyis az élek halmaza igazából csúcspárok halmaza). Ebben a tárgyban csak olyan gráfokkal fogunk foglalkozni, ahol az élek két különböző csúcsot kötnek össze és két csúcs között legfeljebb egy él fut. A csúcsok számát n, az élek számát e betűvel fogjuk jelölni.

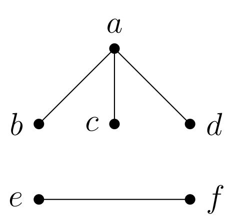
Irányítatlan gráfok például az alábbiak:

G_1:



Itt hat csúcs van, az élek halmaza pedig $E = \{12, 13, 14, 25, 46, 56\}$. Az éleknek nincs irányítása, 12 csak annyit jelent, hogy az 1 és 2 csúcsok össze vannak kötve.

G_2:



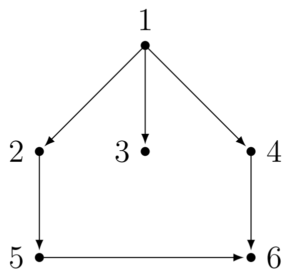
Itt is hat csúcs van, az élek halmaza pedig $E = \{ab, ac, ad, ef\}$. Ez a gráf olyan, hogy nem lehet bármely csúcsából bármely másik csúcsát elérni, az a, b, c, d és az e, f halmazok között nincs összeköttetés.

Egy n csúcsú irányítatlan gráfban az élek minimális száma 0 (lehet, hogy egy él sincsen), maximális száma $\binom{n}{2}$ (amikor minden csúcspár össze van kötve).

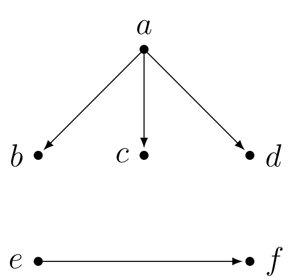
Irányított gráf: a G irányított gráfot is a V csúcshalmaz és E élhalmaz adja meg. A V csúcshalmaz itt is az $1, 2, 3 \ldots, n$ címkékkel megjelölt csúcsokból áll (bár néha a csúcsokat az órán inkább betűkkel fogjuk jelölni), az élek viszont irányítottak, azaz minden él egy csúcsból egy másik csúcsba vezet (vagyis az élek halmaza rendezett csúcspárok halmaza). Irányított gráfok esetén is csak olyan gráfokkal fogunk foglalkozni, ahol az élek két különböző csúcsot kötnek össze és két csúcs között mindegyik irányban legfeljebb egy él fut (az azonban lehetséges, hogy két csúcs között mindkét irányba vezet egy-egy él). A csúcsok számát n, az élek számát itt is e betűvel fogjuk jelölni.

Irányított gráfok például az alábbiak:

G_3:



G_4:



Egy n csúcsú irányított gráfban az élek minimális száma 0 (itt is lehetséges, hogy egy él sincsen), maximális száma $2 \cdot \binom{n}{2} = n(n-1)$ (amikor minden csúcspár össze van kötve mindkét irányban).

Fokszám

Irányítatlan gráfban egy csúcs **fokszáma** a csúcsra illeszkedő élek száma. A v csúcs fokát $d(v)$-vel jelöljük. A fenti G_1 gráfban az 1-es csúcs fokszáma, azaz $d(1) = 3$, a 3-as csúcs fokszáma 1, a többi csúcs fokszáma pedig 2. A G_2 gráfban az a csúcs fokszáma $d(a) = 2$, a többi csúcsé pedig 1.
Irányított gráfban egy v csúcs **ki-foka** azon élek száma, amik a csúcsból indulnak ki (ennek jele $d_{ki}(v)$, a v csúcs **be-foka** pedig (jele $d_{be}(v)$ pedig azon élek száma, amik v-be érkeznek. A fenti G_3 gráfban például $d_{ki}(1) = 3$ és $d_{be}(1) = 0$, míg G_4-ben $d_{ki}(f) = 0$ és $d_{be}(f) = 1$.

Állítás

(a) Irányítatlan gráfban a fokszámok összege $2e$, azaz

$$\sum_{v \in V} d(v) = 2e$$

(b) Irányított gráfban a ki-fokok összege e, azaz

$$\sum_{v \in V} d_{ki}(v) = e$$

(c) Irányított gráfban a be-fokok összege e, azaz

$$\sum_{v \in V} d_{be}(v) = e$$

Bizonyítás

(a) Amikor a fokszámokat összeadjuk, akkor minden élet kétszer számolunk, az él két végpontjánál.

(b) Amikor a ki-fokokat összeadjuk, akkor minden élet egyszer számolunk meg, annál a csúcsnál, ahonnan indul.

(c) Amikor a be-fokokat összeadjuk, akkor minden élet egyszer számolunk meg, annál a csúcsnál, ahova érkezik.

Szomszédossági mátrix

A gráfokat az órán rajzzal fogjuk megadni, de ez csak akkor működik, ha a gráf nem túl nagy és persze ez a módszer akkor sem működik, ha számítógépes programban akarunk a gráfokkal dolgozni. Többféle reprezentáció létezik, amivel a gráfokat egy kódban is meg tudjuk adni, mi ebből az egyiket fogjuk tanulni, a szomszédossági mátrixot. Egy másik lehetséges reprezentáció lenne az éllista, erről egy kicsit a Programozás alapjai tárgyban fognak hallani.

Egyelőre azt az esetet tárgyaljuk csak, amikor az éleknek nincsen súlya, az élsúlyozott szomszédossági mátrixot akkor fogjuk majd bevezetni, amikor az első, élsúlyozott gráfokon futó algoritmust tanuljuk.

Irányítatlan gráf szomszédossági mátrixa: Egy n csúcsú (nem élsúlyozott) irányítatlan gráf A-val jelölt szomszédossági mátrixában az $A[i, j]$ elem (az i-edik sor j-edik oszlopban álló érték) 1, ha az i és j csúcsok között van él, egyébként pedig $A[i, j] = 0$.

A korábban látott G_1 gráf szomszédossági mátrixa

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Vegyük észre, hogy

- a szomszédossági mátrixból könnyen leolvasható az élek száma: a mátrixban szereplő 1-es értékek számának a fele (hiszen minden élet kétszer reprezentálunk)

- a szomszédossági mátrixból a fokszámok is leolvashatók: az i csúcs fokszáma az i-edik sorban szereplő 1-es értékek száma (tehát egy csúcsát fokszámát a mátrixból $O(n)$ lépésben meg tudjuk határozni)
- a szomszédossági mátrix mérete n csúcs esetén n^2, ennyi cellából áll a mátrix
- irányítatlan gráf esetén a mátrix szimmetrikus, azaz $A[i,j] = A[j,i]$, mert pontosan akkor van él i és j között, amikor él van j és i között
- a szomszédossági mátrix főátlójában csupa 0 van, mert az élek két különböző csúcsot kötnek össze, nincs él egyik csúcsból sem saját magába

Irányított gráf szomszédossági mátrixa: Egy n csúcsú (nem élsúlyozott) irányított gráf A-val jelölt szomszédossági mátrixában az $A[i,j]$ elem (az i-edik sor j-edik oszlopban álló érték) 1, ha van él a gráfban az i csúcsból a j csúcsba, egyébként pedig $A[i,j] = 0$.

A korábban látott G_3 gráf szomszédossági mátrixa

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Vegyük észre, hogy

- a szomszédossági mátrixból könnyen leolvasható az élek száma: a mátrixban szereplő 1-es értékek száma (hiszen minden élet egyszer reprezentálunk)
- a szomszédossági mátrixból a ki- és be-fokok is leolvashatók: az i csúcs ki-foka az i-edik sorban szereplő 1-es értékek száma , az i csúcs be-foka pedig az i-edik oszlopban levő 1-ek száma (tehát egy csúcs ki- vagy be-fokát a mátrixból $O(n)$ lépésben meg tudjuk határozni)
- a szomszédossági mátrix mérete n csúcs esetén n^2 az irányított esetben is, ennyi cellából áll a mátrix
- irányított gráf esetén a mátrix általában nem szimmetrikus, azaz $A[i,j]$ és $A[j,i]$ értéke általában nem ugyanaz, mert az, hogy van él i-ből j-be nem ugyanaz, mint hogy van él j-ből i-be
- a szomszédossági mátrix főátlójában irányított esetben is csak 0 van, mert nincs él egyik csúcsból sem saját magába