Algel XII. gyakorlat

Gráfokat feszítünk

2012. április 21.

3. **[Vizsga: 2005. június 23.] Irányítatlan gráf tárolására adjon meg egy adatszerkezetet az alábbi műveletekkel:**
   ÚJCSÚCS(v): a gráfhoz hozzáad egy új csúcsot;
   ÚJÉL(u, v): a már létező u és v csúcsok közé felvesz egy élet;
   VANÚT(u, v): *igen* értéket ad vissza, ha vezet az u és v csúcsok között út, egyébként pedig *nem* értéket.
   Ha a tárolt gráfnak n csúcsa van, akkor mindhárom művelet lépésszáma legyen $O(\log n)$.
   Gyakorlatilag UNIÓ-HOLVAN. Azért indokolni és lépésszámokat számolni nem árt.

4. **[Vizsga: 2010. május 27.] Adott egy $G = (V, E)$ irányítatlan, összefüggő, súlyozott gráf az éllistájával valamint egy $f \in E$ él. Tegyük fel, hogy a gráfban minden él súlya különböző. Adjon $O(|V|+|E|)$ lépésszámú algoritmust annak eldöntésére, hogy van-e olyan minimális feszítőfa G-ben, amely tartalmazza az f élet!**
   Először belátjuk, hogy amennyiben minden él súlya különböző, akkor pontosan 1 minimális súlyú feszítőfa van. Tfh ez nem így van, azaz $\exists F_1, F_2$ feszítőfa, mindkettő súlya minimális, és $\exists e$ él úgy, hogy $e \in F_1, e \notin F_2$. Ekkor e behúzásával egy kör keletkezik F_2-ben, ez a kör legyen C. Ha e súlya kisebb, mint C valamely élének súlya, akkor C-nek ezt az élét e-re cserélve F_2 súlyát csökkenthetnénk, vagyis F_2 nem lehetett minimális. Ha e súlya nagyobb C bármely élének súlyánál, akkor viszont a piros szabály alkalmazható rá, vagyis nem lehetett volna F_1 része.

   Tehát a minimális súlyú feszfa egyértelműsége miatt elég azt ellenőrizni, hogy f része-e ennek a feszfának. Sajnos magát a feszfát túl drága lenne megkeresni, de nem is kell: nézzük meg, hogy alkalmazható-e f-re a piros szabály. Azaz keressünk kört G-ben (pl bejárással) úgy, hogy f-en kívül csak a nála kisebb súlyú éleket használhatjuk. Ha van ilyen kör, akkor f-re alkalmazható a piros szabály, vagyis nem része a minfeszfának, ha pedig nincs, akkor f-re nem alkalmazható a piros szabály, tehát része a minfeszfának. Mivel csak egy bejárást csináltunk (pl mélységi), a lépésszám $O(|V| + |E|)$.

5. **[Vizsga: 2008. június 3.] Éllistával adott a $G = (V, E)$ egyszerű, összefüggő gráf. A gráf élei súlyozottak, a súlyfüggvény $c : E \to \{-1, 1\}$. Adjon algoritmust, ami G-ben $O(|V| + |E|)$ lépésben meghatározza, hogy mennyi a minimális súlya egy olyan részgráfnak, ami G minden pontját tartalmazza és összefüggő.**
   A keresett részgráfba minden -1 súlyú élet be kell válogatni (ha nem tennénk, akkor egy ilyen él bevételével jobbat kapnánk). Mostmár csak az összefüggőséget kell biztosítani. Bejárással határozzuk meg a -1 súlyú élek által feszített részgráf komponenseinek számát, ez legyen k. Ekkor a súly $\sum_{s(e)=-1} -1 + k - 1$, hiszen a komponenseket $k - 1$ darab 1 súlyú éllel kell és elég összekötni (ennyi elég és lehetséges G öf miatt; ha kevesebb, akkor meg marad külön komponens). A lépészám a bejárásé: $O(|V| + |E|)$.

6. **[Vizsga: 2007. június 5.] Útépítéskor a környéken sok helyen felszedték a járdát. Az építők 1-től n-ig megszámozták a fontos pontokat (kapualj, útkereszteződés, stb.). A környék állapotát két $n \times n$ táblázat írja le. A J táblázatban $J[i, j] = 1$, ha az i és j pontok az utcán szomszédosak és megmaradt az ezeket összekötő részen a járda, egyébként az érték 0. A P táblázat az ideiglenesen elhelyezhető pallókat írja le: ha az i és j pontok összeköthetőek egy pallóval, akkor $P[i, j]$ ennek a pallónak a költsége. Amennyiben a két pont nem köthető össze egy pallóval, akkor a táblázatban $*$ szerepel. (Minden palló pontosan két pontot érint.) Szeretnénk biztosítani, hogy**

mindenhonnan mindenhova el tudjunk jutni (hol járdán hol pallón haladva). Az építők célja, hogy úgy válasszák meg a pallók helyét, hogy minél kevesebb pallót kelljen használniuk, és ezen belül a pallók értékeinek összege minimális legyen. Írjon le egy algoritmust, ami $O(n^2)$ lépésben javasol egy ilyen elhelyezést. (Egy pontra tetszőlegesen sok palló illeszkedhet, és a gyalogosok az egy pontra illeszkedő pallók bármelyikéről bármelyikére át tudnak lépni.)

Először megmaradt járdákat használva öf kompnensek keresés, majd a komponensek között min ktg feszfa. Helyesség és lépésszám bizonyítandó!

7. **Mátrixával adott egy G irányítatlan súlyozott gráf. Adott még a G-nek egy F minimális súlyú feszítőfája, és az F-nek egy f éle. Adjon $O(n^2)$ lépésszámú algoritmust, ami meghatározza, hogy az f él súlyát meddig lehet úgy felemelni, hogy az F a gráf minimális feszítőfája maradjon.**

   f elhagyásával F két komponensre esik, ezeket egy bejárással meg tudjuk keresni (és minden csúcsnál megjegyezzük, hogy melyik komponensbe tartozik). f-et pontosan a két komponens között futó élek tudják helyettesíteni, így közülük a minimális súlyú súlyáig tudjuk f-et növelni. Tehát G (F-en kívüli) összes élén végigmegyünk, és a két komponens között futók súlyának minimumát megkeressük. Lépésszám: bejárás és éleken végigmenés: $O(n^2 + n^2) = O(n^2)$.

8. **[Vizsga: 2008. május 27.] Éllistával adott egy egyszerű, összefüggő, súlyozott irányítatlan $G = (V, E)$ gráf amiben nincs két egyforma súlyú él. Adjon $O(|V| \cdot |E|)$ lépésszámú algoritmust, ami megadja a gráfban a második legkisebb súlyú feszítőfát.**

   Vázlatosan: a legkisebb súlyú feszítőfától a különböző élsúlyok miatt pontosan egy élben fog eltérni (biz. vázlat: tfh legalább kettőben, így az egyiket a minfeszfa megfelelő élére cserélve kapnánk egy kisebb súlyút ami még nem min, tehát a jelöltünk nem lehetett a második legkisebb). Így keresünk egy feszítőfát, és abból egyenként az összes $(n-1)$ élet kék helyett pirosra színezzük, és keresünk helyette egy másik kék élet. A kapott $n-1$ érték közül a legkisebbet vesszük. Lépésszám: $O(|E| \log |E| + |V| \cdot |E|) = O(|V| \cdot |E|)$, feszfakeresés és utána $n-1$ kék szabály.

9. **Legyen adva egy (egyszerű, irányítatlan, összefüggő) n csúcsú G gráf éllistával, az élek súlyozásával együtt. Tegyük fel, hogy a G-ből a v_1 csúcs, valamint a v_1-re illeszkedő élek elhagyásával keletkező G' gráf még mindig összefüggő, és adott a G' egy minimális költségű feszítőfája. Adjunk $O(n \log n)$ futási idejű algoritmust a G gráf egy minimális költségű feszítőfájának elkészítésére!**

   A piros-kék algoritmus a G'-ben pirosra színezett éleket G-ben is pirosra színezné (a kékeket nem biztos, hogy kékre! – szorgalmi feladatként mutassunk egy példát, ahol egy G'-beli kék él G-ben piros lesz!). Ezért elég, ha G' feszítőfájához vesszük hozzá v_1-et, és ebben a gráfban keresünk minktg feszfát. Ez Prim vagy Kruskal algoritmussal éllistás megadásban $O(|E| \log |E|)$, és ebben az esetben legfeljebb $n - 1 + n$ élünk van (feszítőfa és új élek), így a költség $O(n \log n)$ lesz.

10. **Hány éle van az n pontú egyszerű összefüggő gráfnak, ha pontosan 3 különböző feszítőfája van?**

    Legalább n, hiszen $n-1$ esetén fa lenne, így ekkor csak egy feszítőfája lehetne. Ha van benne egy k hosszú kör, akkor a kör bármely élét elhagyva a maradék kiegészíthető feszítőfává, így ekkor legalább k darab van neki. Vagyis csak 3 hosszú kör(ök) lehet(nek) a gráfban. Ha több 3 hosszú kör is van, akkor belőlük függetlenül el lehet hagyni éleket, így a feszítőfák száma 3 egész számú többszöröse lesz. Vagyis a gráfban pontosan egy darab, 3 hosszú kör van (azaz egy háromszög, aminek a csúcsairól fák lóghatnak le), így egy ilyen gráfnak legfeljebb n éle lehet. Fentieket összerakva (legalább és legfeljebb n) pontosan n éle van.

11. **Éllistával adott egy összefüggő, egyszerű, irányítatlan, n csúcsú, e élű G gráf csupa különböző élsúllyal. Adjunk egy olyan $O(e)$ költségű algoritmust, ami a G gráf**

egy minimális feszítőfájának legalább $\frac{2}{3}n$ élét előállítja! (Azaz egy olyan élhalmazt keresünk, ami biztosan része egy minimális költségű feszítőfának.)
Borůvka algoritmusának első két menetét futtatjuk (első menet után legalább $\left\lfloor \frac{n}{2} \right\rfloor$ él van meg, második után legalább $\left\lfloor \frac{3n}{4} \right\rfloor$).

12. **Bizonyítsuk be, hogy a piros-kék algoritmus akkor is helyesen működik, ha egy feszítőfa költségét az élsúlyok összege helyett az élsúlyok maximumával definiáljuk! (Ez az állítás vizsgán bizonyítás nélkül is felhasználható, ha úgy adódik.)**
A piros-kék algoritmus helyességének bizonyításában ezzel a célfüggvénnyel is minden állítás érvényes lesz (nyilván végig kell nézni).