Algel XII. gyakorlat

Feszítsünk gráfokat, olcsón!

2009. április 28/30.

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

5. **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!). 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.

6. **[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 $k-1$ 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: $O(|V| + |E|)$.

7. **[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 (ezt meg kell indokolni!). Így keresünk egy feszítőfát, és abból egyenként az összes $(n-1)$ élet kék helyett pirosra színezzük, és keresünk egy 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.

8. **[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!

9. **[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.