Algel XIV. gyakorlat

Egészértékű programozás, *közeledik* a félév vége, könnyes búcsú

2011. május 9.

3. **[Vizsga: 2009. június 4.] Egy adott egyszerű, irányítatlan gráfban maximális méretű teljes részgráfot akarunk találni. Írja le ezt a problémát egy egész értékű programozási feladatként! (A kapott egész értékű programozási feladatot nem kell megoldani.)**
   Minden ponthoz felveszünk egy x_i (bináris) változót (binárisságot biztosító korlátok: (3,4,5)), ami 1, ha a v_i pont a keresett részgráfban található, egyébként 0. A bevett pontok számát akarjuk maximalizálni, ami megfelel a változók összegének maximalizálásának (1). Két pont nem lehet benne egy teljes részgráfban, így ha összekötetlenek, egyszerre nem lehet őket kiválasztani (2).

$$\max \sum_{i=1}^{n} x_i \tag{1}$$

feltéve, hogy

$$\begin{aligned}
x_i + x_j &\leq 1 \quad \forall i,j : (v_i, v_j) \notin E && (2)\\
x_i &\leq 1 \quad \forall i = 1 \dots n && (3)\\
x_i &\geq 0 \quad \forall i = 1 \dots n && (4)\\
x_i &\in \mathbb{Z} \quad \forall i = 1 \dots n && (5)
\end{aligned}$$

4. **[Vizsga: 2010. június 3.] Fogalmazza meg egész értékű programozási feladatként az alábbi problémát! Egy adott $G = (V, E)$ irányítatlan egyszerű gráfban keresünk olyan maximális méretű $D \subseteq V$ csúcshalmazt, melyre teljesül, hogy minden $x \in V$ csúcsnak legfeljebb 2 szomszédja van a D halmazban!**
   Minden ponthoz felveszünk egy x_i (bináris) változót (binárisságot biztosító korlátok: (3,4,5)), ami 1, ha a v_i pont a keresett csúcshalmazban található, egyébként 0. A bevett pontok számát akarjuk maximalizálni, ami megfelel a változók összegének maximalizálásának (1). Minden csúcsra igaz, hogy a szomszédai közül legfeljebb kettő lehet kiválasztva, azaz a szomszédoknak megfelelő változók összege nem lehet 2-nél nagyobb (2).

$$\max \sum_{i=1}^{n} x_i \tag{1}$$

feltéve, hogy

$$\begin{aligned}
\sum_{\forall j:(v_i,v_j)\in E} x_j &\leq 2 \quad \forall i = 1 \dots n && (2)\\
x_i &\leq 1 \quad \forall i = 1 \dots n && (3)\\
x_i &\geq 0 \quad \forall i = 1 \dots n && (4)\\
x_i &\in \mathbb{Z} \quad \forall i = 1 \dots n && (5)
\end{aligned}$$

5. **[Vizsga: 2010. június 17.] Adott egy $G = (V, E)$ egyszerű, irányítatlan gráf. Egy olyan $W \subseteq V$ halmazt keresünk, amely a lehető legtöbb csúcsból áll és teljesül rá, hogy a gráfban bármely 2 független él 4 végpontjából W legfeljebb 2 pontot tartalmaz.**

Hogyan lehet ezt a problémát egészértékű programozási feladatként felírni? (A kapott EP feladatot nem kell megoldani!)
Minden ponthoz felveszünk egy x_i (bináris) változót (binárisságot biztosító korlátok: (3,4,5)), ami 1, ha a v_i pont a keresett csúcshalmazban található, egyébként 0. A bevett pontok számát akarjuk maximalizálni, ami megfelel a változók összegének maximalizálásának (1). Minden független élpárra igaz, hogy a hozzájuk tartozó 4 csúcs közül legfeljebb kettő lehet kiválasztva, azaz minden ilyen 4 csúcsnak megfelelő változók összege nem lehet 2-nél nagyobb (2).

$$\max \sum_{i=1}^{n} x_i \tag{1}$$

feltéve, hogy

$$\begin{aligned}
x_i + x_j + x_k + x_l &\leq 2 \quad \forall e_a, e_b : e_a = (v_i, v_j), e_b = (v_k, v_l), \text{egymástól ftln élek} && (2)\\
x_i &\leq 1 \quad \forall i = 1 \dots n && (3)\\
x_i &\geq 0 \quad \forall i = 1 \dots n && (4)\\
x_i &\in \mathbb{Z} \quad \forall i = 1 \dots n && (5)
\end{aligned}$$

Megjegyzés: nem tartozik a feladathoz, de érdemes belegondolni, hogy a páronként független élek megkeresése nyilván nem okoz gondot, hiszen végigmegyünk az összes lehetséges élpáron ($O(e^2)$), és a páronként függetlenekre felvesszük a korlátot.

6. **[Vizsga: 2009. június 17.] A Ládapakolás problémának tekintsük azt a speciális esetét, amikor az n tárgy mindegyike vagy a vagy b méretű, ahol $0 < a < b < 1$ és $a + b = 1$. Adjon ennek megoldására $O(n)$ lépésszámú algoritmust!**
Algoritmus: ameddig lehet, párosával rakjuk az a és b méretű elemeket ládába, a maradékot pedig a FirstFit algoritmussal rendezzük el. Ez ügyesen implementálva ebben a spec esetben triviálisan $O(n)$.
Helyesség: az a méretű elemek darabszáma legyen A, a b méretűeké B. Ha $B \geq A$, akkor az algoritmus B ládát fog felhasználni ($OPT \leq B$), és mivel $b > 1/2$, mindegyik külön ládába kell, hogy kerüljön, így legalább B ládára szükség is van ($OPT \geq B$). Ha $A > B$, akkor két dolgot bizonyítunk.
Az egyik, hogy mindig létezik olyan optimális elrendezés, hogy minden b méretű elem mellett szerepel egy a méretű is. Tfh egy olyan optimális elrendezésünk van, ahol legalább az egyik b-nek nincs párja. Ekkor egy csak a-t tartalmazó ládából egy a-t át tudunk rakni ide úgy, hogy a szükséges ládák számát nem növeljük. Ezt addig csinálhatjuk, amíg a kívánt elrendezéshez nem jutunk, és közben a ládák számát egyszer sem növeltük. Ez alapján elég az olyan elrendezéseket vizsgálni, ahol minden b párosítva van egy a-val (azaz amilyen elrendezéseket a mi algoritmusunk is ad), és csak az a-k elrendezésére kell koncentrálni.
Tfh nem a FirstFit szerinti módon vannak a maradék a-k bepakolva. Ekkor az általánosság megsértése nélkül feltételezhetjük, hogy a bepakolt elemek darabszáma szerint nemnövekvően vannak rendezve a ládák. A feltételezés szerint az utolsó ládán kívül is van olyan, ahova még beférne egy a méretű elem (különben pont a FirstFit szerinti elrendezésünk lenne). A legutolsó ládából ekkor egy a-t átpakolhatunk ide anélkül, hogy a ládák számát növelnénk. Ezt egészen addig csinálhatjuk, amíg nem a kívánt formában van az elrendezés, és a ládák száma egyszer sem nőtt. Azaz mindig létezik olyan optimális megoldás, mint amilyet a mi algoritmusunk ad.
Ezt a bizonyítást természetesen le lehetne írni sokkal rövidebben is, de a jó érthetőség és a precizitás miatt választottam ezt.

7. **[Vizsga: 2009. május 28.] Egy csomagküldő szolgálatnál csupa egyforma dobozokba pakolják az elküldendő árut. Céljuk az, hogy a ládákban maradó üres helyet kitöltő anyagból minél kevesebbre legyen szükség. Igaz-e, hogy a Ládapakolásra megismert**

First Fit eljárás erre a problémára is egy 2-közelítő algoritmus?
Ellenpélda: dobozméret: 1, elemek: 0.6, 0.3, 0.7, 0.4. Optimális megoldás: $[0.6, 0.4], [0.7, 0.3]$, $OPT = 0$ (mert 0 hely marad ki). A FirstFit által adott megoldás: $[0.6, 0.3], [0.7], [0.4]$, a kimaradó hely 1. Ha 2-közelítő lenne az algoritmus, akkor $1 \leq 2OPT = 0$ teljesülne, ami nyilván nincs így. Vagyis a válasz nem.

8. **[Vizsga: 2006. június 12.] Van n fájlunk, az i-edik fájl hosszát jelölje h_i. Tegyük fel, hogy a fájlok hosszuk szerint nem csökkenő sorrendben követik egymást, azaz $0 < h_1 \leq h_2 \leq \cdots \leq h_n$. Mentéskor két egyforma méretű lemez áll rendelkezésünkre. A mentésnek sorban kell történnie, előbb az első fájlról kell megmondani, melyik lemezre kerüljön, azután a másodikról, stb. (Fájlokat szétvágni nem szabad, minden fájl teljes egészében kerül az egyik vagy a másik lemezre.) Amikor a soron következő fájl már egyik lemezre sem fér rá, akkor abbahagyjuk az eljárást. Egy ilyen eljárás optimális, ha a lehető legtöbb fájlt lehet segítségével kimenteni.**
Mutassa meg, hogy az a mohó eljárás, amikor a következő fájlt oda tesszük, ahol több hely van, nem feltétlenül optimális. Legfeljebb hány fájllal fogunk kevesebbet kimenteni ezzel a mohó eljárással az optimális (szintén sorrendben mentő) megoldáshoz képest?
Az, hogy a mohó algoritmus nem optimális, egy ellenpéldával könnyen bizonyítható: a szalag mérete legyen 2, az érkező fájlok mérete 1,1,2. Az optimális megoldásban mind a 3 fájlt ki tudjuk írni (az egyik szalagon 1,1, a másikon 2), a mohó csak a két 1-est írja ki, a két szalagra.

Némi gondolkodás után megsejtjük, hogy a mohó algoritmus mindig csak legfeljebb 1-gyel kevesebb fájlt ír ki, mint amennyit az optimális kiírás szerint lehetne. Ezt bizonyítjuk be. A szalagok mérete legyen S. Tfh van egy olyan feladatunk, amiben legalább 2 fájllal többet lehet kiírni, mint amit a mohó változat ad. Ezek közül az első kettő hossza legyen h_i és h_{i+1}, a feltételből tudjuk, hogy

$$h_i \leq h_{i+1} \tag{1}$$

Az egyik szalagon W_1 méret foglalt, a másikon W_2. Ha a maradék $S - W_1$ vagy $S - W_2$ helyre beférne h_i, akkor azt a mohó algoritmus berakta volna. Vagyis

$$S - W_1 < h_i \tag{2}$$

$$S - W_2 < h_i \tag{3}$$

Az optimális megoldásban W_1 és W_2 mennyiségű adat biztosan szerepel valamilyen elrendezésben, és ezen kívül h_i és h_{i+1} is még befér. Azaz az összes elérhető és felhasznált adatmennyiséget felírva ebben az esetben

$$2S \geq W_1 + W_2 + h_i + h_{i+1} \geq W_1 + W_2 + 2h_i \tag{4}$$

ahol a második egyenlőtlenség (1)-ből következik. Ezt (2) és (3) segítségével tovább írva (figyeljünk, hogy hogy hol van $\geq$, és hol van $>$!):

$$2S \geq W_1 + W_2 + 2h_i > W_1 + W_2 + (S - W_1) + (S - W_2) = 2S \tag{5}$$

ami nyilvánvalóan lehetetlen. Így nem lehet, hogy a mohó algoritmus által kiírt fájloknál legalább 2-vel több kiírható lenne.

9. **Gondolkozzunk el a félévben tanultakon, fogalmazzunk meg kérdéseket, stb!**
De tényleg!

10. **Oldj meg sok feladatot a vizsgára, tanulj sokat, ha valami nem világos, akkor esetleg a gyakvezéredet is megkérdezheted a `drotos@cs.bme.hu` címen!**
Május 28. és június 4. között nem biztos, hogy gyakran tudok (vagy egyáltalán tudok) emailt olvasni, ezt vegyétek figyelembe!

11. **???**

12. **Profit! (80 pont körüli vizsga!)**