NP-teljesség; Egészértékű programozás

Algoritmuselmélet

2019. tavasz

10. gyakorlat

Tétel. Az alábbi problémák NP-teljesek.

3DH
Bemenet: U_1, U_2, U_3 t-elemű halmazok és $S \subseteq U_1 \times U_2 \times U_3$.
Kérdés: létezik-e olyan t-elemű $S' \subseteq S$ részhalmaz, ami minden $U_1 \cup U_2 \cup U_3$-beli pontot lefed?

X3C
Bemenet: U véges alaphalmaz és $X_1, \ldots, X_k \subseteq$ 3-elemű részhalmazok.
Kérdés: kiválasztható-e ezekből páronként diszjunkt halmazok úgy, hogy ezek lefedjék U-t?

Egészértékű programozás alapfeladata (optimalizálási változat)

Adott $x_1, \ldots, x_m$ változókat tartalmazó lineáris egyenlőtlenségrendszer egész megoldásai között mekkora $\max(c_1 x_1 + \ldots + c_m x_m)$?

c-közelítő algoritmus

Legyen adott egy optimalizálási probléma és jelölje OPT az optimális megoldás értékét. Egy algoritmust c-közelítő algoritmusnak nevezünk, ha az adott optimalizálási probléma minden bemenetére polinomidőben ad egy olyan megoldást, melynek értéke maximalizálási probléma esetén legalább $\frac{1}{c} \cdot \mathrm{OPT}$, minimalizálási probléma esetén pedig legfeljebb $c \cdot \mathrm{OPT}$.

1. Igazoljuk, hogy léteznek az alábbi Karp-redukciók!

$$\mathrm{HAM} \prec 10\mathrm{SZÍN} \prec \mathrm{X3C} \prec \mathrm{HAM}$$

2. Egy programozási versenyen F fős csapatok vehetnek részt. Az érdeklődő hallgatók száma H, ők meghatározták a belőlük formálható összes lehetséges F fős csapatot, figyelembe véve, hogy ki kivel hajlandó egy csapatban lenni. Feltehetjük, hogy minden hallgató benne van legalább egy lehetséges csapatban, de egy személy többen is szerepelhet. A versenyen ténylegesen elinduló csapatokban természetesen nem lehet közös tag. Kérdés, hogy a megadott C lehetséges csapat közül ki tudjuk-e választani a versenyen ténylegesen induló csapatokat úgy, hogy minden hallgató részt vehessen a versenyen valamelyik csapat tagjaként. P-beli vagy NP-teljes a probléma?

3. Tegyük fel, hogy van egy eljárásunk, ami tetszőleges Boole-formuláról polinomidőben eldönti, hogy eleme-e a SAT nyelvnek. Hogyan lehet ezt polinomidőben megtalálni egy adott $\varphi(x_1, x_2, \ldots, x_n)$ formulához a változóknak egy olyan értékelését, amelyet ha a φ-be helyettesítünk, akkor a formula értéke igaz lesz?

4. Tegyük fel, hogy van egy $\mathcal{E}$ eljárásunk, ami tetszőleges egyszerű (súlyozatlan), irányított gráfban meghatározza, hogy mennyi a benne levő leghosszabb irányított kör hossza. Adott G irányított gráfra az $\mathcal{E}$ eljárás segítségével találjuk meg, hogy mely élekből áll a G-beli leghosszabb kör (ha több ugyanolyan hosszú kör is van, akkor elég az egyiknek az éleit meghatározni)! Az algoritmus során az $\mathcal{E}$ eljárás meghívásainak száma és a hívásokon kívül végrehajtott többi lépés száma is legyen polinomiális n-ben, ahol n a G csúcsainak számát jelöli.

5. Tegyük fel, hogy MAXFTL $\in$ P. Ezt felhasználva mutassuk meg, hogyan lehet tetszőleges G egyszerű gráfban polinomidőben megtalálni
   (a) a legnagyobb független ponthalmaz méretét;
   (b) magát egy legnagyobb független ponthalmazt!
6. Tegyük fel, hogy van egy eljárásunk, ami egy tetszőleges n csúcsú gráfról polinomidőben megmondja, hogy az kiszínezhető-e három színnel. Hogyan lehet ezt felhasználva polinomidőben megtalálni egy adott G gráfnak egy 3 színnel színezését (ha van ilyen egyáltalán)?
7. A $G(V, E)$ egyszerű gráf élei súlyozottak. Olyan $X \subseteq E$ maximális súlyú élhalmazt akarunk kiválasztani, hogy minden csúcsra legfeljebb 3 darab X-beli él illeszkedjen. Írjuk fel egészértékű programozási feladatként ezt a problémát!
8. Adott egy $G = (V, E)$ egyszerű irányítatlan gráf. Minden éléhez egy súlyt akarunk rendelni, a súlyok mindegyike a $0, 1, \ldots, 10$ egész számok közül kerülhet ki. Célunk, hogy G-ben az élek súlyainak összege maximális legyen, de egyik csúcsnál se legyen a rá illeszkedő élek súlyainak összege 15-nél nagyobb. Írjuk fel egészértékű programozási feladatként ezt a problémát!
9. Írjuk fel egészértékű programozási feladatként
   (a) a 3DH probléma optimalizálási változatát;
   (b) a minimális lefogó ponthalmaz keresését egyszerű, irányítatlan gráfban.
10. Adott az $a_1, a_2, \ldots, a_n$ számsorozat. Ebből néhány tagot akarunk úgy kiválasztani, hogy ne legyen közöttük két szomszédos eleme a sorozatnak, és a kiválaszott elemek négyzetösszege maximális legyen. Írjuk fel egészértékű programozási feladatként ezt a problémát!
11. Egy reklámkampányhoz plakátokat akarunk kirakni. A városban elérhető n plakáthely mindegyikéhez adott, hogy ott várhatóan hányan látják majd, legyenek ezek a számok $k_1, k_2, \ldots, k_n$. Adott továbbá, hogy az i-edik és j-edik hely t_{ij} távolságra van egymástól. Összesen legfeljebb p plakátot tennénk ki és ezekhez úgy szeretnénk a $p \leq n$ helyet kiválasztani, hogy ezek közül bármely kettő távolsága legalább T legyen és várhatóan minél többen lássák őket. (Feltehetjük, hogy a megfelelő k_i értékek összeadódnak, továbbá, hogy a szereplő k_i és t_{ij} számok egészek.) Írjuk fel egészértékű programozási feladatként ezt a problémát!
12. Az UTAZÓÜGYNÖK problémára tekintsük a következő mohó algoritmust: tetszőleges csúcsból indulunk és minden alkalommal a legkisebb olyan súlyú élen megyünk tovább, ami új csúcsba visz. Ha már minden csúcsban jártunk, akkor egy élen visszatérünk a kezdő csúcsba. Tetszőleges adott $c \geq 1$ konstansra adjunk meg olyan gráfot, ami mutatja, hogy ez nem egy c-közelítő algoritmus.