SzA X. gyakorlat

Rendezkedünk, továbbá bevezetés a T-betűs szó világába

2010. november 10/11.

1. **Gondolkozzunk el az NP-beliség, $coNP$-beliség és P-beliség fogalmakon!** De tényleg!

2. **A valós számokból álló $a_1^2 \dots a_n^2$ sorozat egy darabig nő, utána csökken. Adjunk egy legfeljebb $c \cdot n$ összehasonlítást használó algoritmust, ami rendezi az $a_1, \dots a_n$ sorozat!**
   Megkeressük, hogy a sorozat hol vált át csökkenőbe (érelemszerűen a sorozat elemeinek négyzeteivel foglalkozunk). Ez n lépésből biztos megvan. Utána összefésüljük a két növekvő és csökkenő sorozatot (még mindig a négyzetüket nézzük), ez n lépés, és lett egy négyzetek szerint növekvő sorozatunk. A pozitívak eleve jól vannak benne, a negatívak pont fordított sorrendben. Egy végigolvasással a negatívakat a sorozat elejére rakjuk, fordított sorrendben, ez ismét n lépés. Összesen tehát $3n$ lépésben megcsináltuk.

3. **Adottak a $p_0 = (0,0), p_1 = (x_1, y_1), \dots p_n = (x_n, y_n), p_{n+1} = (100, 0)$ pontok a síkban ($n \geq 1$) úgy, hogy $1 \leq i \leq n$ esetén x_i és y_i racionális számok, $0 < x_i < 100$, és semelyik három pont nem esik egy egyenesbe. Egyenes szakaszokkal akarjuk ezeket a pontokat valamilyen sorrendben összekötni úgy, hogy egy $n+2$ csúcsú zárt töröttvonalat kapjunk, amiben a behúzott szakaszok nem metszik egymást. Adjunk egy legfeljebb $c \cdot n \log n$ lépést használó algoritmust annak meghatározására, melyik pontot melyikkel kössük össze!**
   Vegyük észre, hogy ha „balról jobbra” az $y > 0$ pontokon megyünk, „jobbról balra” pedig az $y < 0$ pontokon, akkor soha nem fogunk metszést csinálni. A „balról jobbra” és „jobbról balra” pedig pont az x koordináta szerinti növekvő/csökkenő haladást jelenti, azaz egy rendezés után pont meg is tudjuk tenni. Rendezni pedig tudunk $c \cdot n \log n$ lépésben, pl. összefésüléses rendezéssel. (Utána persze még végig kell menni a pontokon az összekötéshez, de az csak n lépés).

4. **Szeretnénk n db SZA-hallgató ZH-ereményeit (csak az összpontszámot) növekvő sorrendben felsorolni. Adjunk erre $c \cdot n$ lépést felhasználó algoritmust!**
   Ládarendezéssel csinálhatjuk, ugyanis 0 és 60 közötti egész számokat kell rendezni.

5. **Be tudjuk-e bizonyítani a következő problémák P, NP és $coNP$-beliségét? Az input egy $G(V, E)$ gráf ($|V| = n, |E| = e$).**

   (a) **Van-e G-ben legalább k hosszú kör? (k az input része.)**
   $\in NP$: Tanú: egy legalább k hosszú kör. Tanú mérete: legfeljebb n (a kör pontjai), ami polinomiális. Ellenőrzés: a megadott pontokon végighaladva megnézzük, hogy tényleg kört alkotnak-e, és tényleg legalább k db-e; polinomiális.

   (b) **Kiszínezhetők-e G pontjai 2 színnel úgy, hogy legfeljebb 2 él kivételével minden él végpontjai különböző színűek?**
   $\in P$: Elhagyunk az összes lehetséges módon két élet, és megnézzük, hogy a

maradék gráf páros-e. Azaz $\binom{n}{2} \cdot poli \approx n^2 \cdot poli$, azaz polinomiális.
$\in NP$: következik $\in P$-ből.
$\in coNP$: következik $\in P$-ből.

(c) **Kiszínezhető-e G 4 színnel?**
$\in NP$: Tanú: egy jó színezés. Tanú mérete: minden ponthoz egy szám (az adott pont színe), azaz polinomiális. Ellenőrzés: minden élre megnézzük, hogy a végpontjaik különböző színűek-e, legfeljebb $c \cdot n^2$ lépés (élek száma), azaz polinomiális.

(d) **Van-e G-ben egy legalább 15 pontú teljes részgráf?**
$\in P$: Az összes $\binom{n}{15} \approx n^15$ részgráfot megnézzük, hogy teljes-e.
$\in NP$: következik $\in P$-ből.
$\in coNP$: következik $\in P$-ből.

(e) **Van-e G-ben egy legalább k pontú teljes részgráf? (k az input része.)**
$\in NP$: Tanú: egy k pontú teljes részgráf. Tanú mérete: legfeljebb n pont, azaz polinomiális. Ellenőrzés: megnézzük, hogy a tanú által kijelölt pontok közül mindegyik össze van-e kötve mindegyikkel, legfeljebb $c \cdot n^2$ lépés (élek száma), azaz polinomiális.

(f) **Síkbarajzolható-e G?**
$\in P$: bizonyítás nélkül kimondtuk.
$\in NP$: következik $\in P$-ből. (Egyébként tanú pl. egy jó lerajzolás.)
$\in coNP$: következik $\in P$-ből. (Egyébként tanú pl. egy Kuratowski-gráffal topologikusan izomorf részgráf.)

(g) **Teljesül-e az Ore-feltétel?**
$\in P$: Az összes összekötetlen pontpárt megvizsgáljuk, és ellenőrizzük a fokszámaik összegét. Legfeljebb $c \cdot n^2$, azaz polinomiális.
$\in NP$: következik $\in P$-ből.
$\in coNP$: következik $\in P$-ből.

(h) **Van-e G-ben legfeljebb S súlyú (egyszerű) út? (S az input része.)**
$\in NP$: Tanú: egy ilyen út. Tanú mérete: az út pontjai vannak megadva, így legfeljebb n, azaz polinomiális. Ellenőrzés: végigmegyünk az úton, összeadjuk a súlyokat, közben figyeljük, hogy tényleg megvannak-e a szükséges élek; legfeljebb n lépés, polinomiális.

(i) **Van-e G-ben olyan feszítőfa, amelyben a maximális fokszám legfeljebb 2?**
$\in NP$: Tanú: egy ilyen feszítőfa. Tanú mérete: pl. Prüfer-kóddal megadás esetén legfeljebb n, azaz polinomiális. Ellenőrzés: végignézzük, hogy tényleg feszítőfa-e, és a fokszámokat is figyeljük, legfeljebb n lépés, polinomiális.

(j) **Van-e G-ben Euler-kör?**
$\in P$: Az ismert tétel alapján a fokszámokat kell végignézni, hogy mindegyik páros-e, és egy összefüggőséget vizsgálni; mindkettő polinomiális.
$\in NP$: következik $\in P$-ből. (Egyébként tanú: egy Euler-kör.)
$\in coNP$: következik $\in P$-ből. (Egyébként tanú: egy páratlan fokszámú pont vagy két komponens, ami a nem összefüggőséget bizonyítja.)

6. **A G irányítatlan gráf minden x pontjához tartozik egy $s(x)$ súly. Célunk, hogy olyan feszítőfát találjunk a gráfban, amiben a levelekhez**

tartozó súlyok összege minimális. Fogalmazzuk meg a feladathoz tartozó eldöntési problémát, és bizonyítsuk be, hogy NP-beli!
Az eldöntési probléma:

$$\pi = \{(G, k) \mid G \text{ irányítatlan csúcssúlyozott teljes gráf, amiben van legfeljebb } k \text{ levélsúlyú feszítő}$$

$\in NP$, mert tanú: egy ilyen feszítőfa. Méret: pl. Prüfer-kóddal megadva legfeljebb n, azaz polinom. Ellenőrzés: végignézzük, hogy feszítőfa-e, és összeadjuk a levelek súlyának összegét; ez az egész megvan polinom időben.

7. **Tegyük fel, hogy van egy algoritmusunk, ami polinom időben megmondja, hogy adott G gráf kiszínezhető-e legfeljebb k db színnel! (Vagyis input: G és k; output: igen/nem). Hogy tudnánk ennek segítségével polinom időben meghatározni $\chi(G)$-t?**
Bináris kereséssel. Rákérdezünk, hogy 1 színnel színezhető-e, Δ színnel színezhető-e, $\Delta/2$-vel, stb. Azaz $\log n$-szer egy polinomiális kérdés, ami még mindig polinomiális.

---

8. **Van 3 algoritmusunk, amelyek n méretű input esetén rendre n^2, n^6 és 2^n lépés alatt végeznek a feladattal. Tegyük fel, hogy egy óra alatt tudunk mindegyikkel megoldani egy n méretű feladatot a számítógépünkön. Ha veszünk egy 100-szor olyan gyors gépet, melyik algoritmussal mekkora feladatot tudunk megoldani ugyanúgy 1 óra alatt?**
n^2: ugyanannyi időnk van mindkét esetben, tehát $n_1^2 = 100n^2$, amiből gyökvonással $n_1 = 10n$, vagyis 10-szer akkorát.
n^6: ugyanígy, $n_1^6 = 100n^6$, amiből $n_1 = \sqrt[6]{100}n \approx 2.15$, vagyis alig több, mint kétszer akkorát.
2^n: a felírás ugyanez, $2^{n_1} = 100 \cdot 2^n$, amiből logaritmussal és ennek azonosságaival $n_1 = n + \log 100 \approx n + 6.64$, azaz nem szorzódik a méret, hanem csak kb 6.64-gyel nagyobb méretű inputra fog ugyanennyi idő alatt lefutni.

9. **Legalább hány összehasonlítás kell ahhoz, hogy egy n elemű tömbből egy olyan tagot találjunk meg, ami a tömb 10 legkisebb eleme közé tartozik?**
$n-10$ összehasonlításnál kevesebb esetén az összehasonlítottsági gráfnak legalább 11 komponense lesz, ami túl sok. $n-10$ elég is, azaz 9 elemet kidobunk, a maradékban meg minimumot keresünk.

10. **Adottak a sík egész koordinátájú $P_1 = (x_1, y_1), \ldots, P_n = (x_n, y_n)$ koordinátájú pontjai. Javasoljunk egy legfeljebb $c \cdot n$ lépésszámű módszert olyan $P_i \neq P_j$ pontok kiválasztására, amelyeken átmenő egyenes által meghatározott félsíkok közül az egyik tartalmazza az összes pontot!**
Minimális y kooridántájú pont lesz az egyik, ami meghatározza, a másik pedig az y_{min}-es pontból abszolútértékben legnagyobb meredekségű. Ha ezeken kívül esne egy pont, akkor annak a meredeksége abszolútértékben nagyobb lenne. Meredekséget számolni két pont ismeretében konstans, így a lépésszám két minimumkeresés, azaz $2 \cdot n$.

11. **Adjunk $c \cdot n$ lépésszámú algoritmust n olyan egész számból álló sorozat rendezésére, melynek elemei az**

    (a) $\{1, \ldots, 3n\}$ **tartományba esnek!** Ládarendezés, $3n$ ládával, $n + 3n = c \cdot n$.

(b) $\{1, \ldots, n^7 - 1\}$ **tartományba esnek!** n-es számrendszerben felírjuk a számokat (ez darabonként 7 osztás, vagyis konstans), utána számjegyenként radix (legfeljebb 7 jegyű számok). (A radix rendezés nem lesz ZH-n.)

12. **Tegyük fel, hogy van egy algoritmusunk, ami polinom időben megmondja, hogy adott G gráf kiszínezhető-e legfeljebb k db színnel! A fentiek értelmében azt is megtudhatjuk polinom időben, hogy mennyi $\chi(G)$. Hogyan tudnánk kiszínezni polinom időben a gráfot $\chi(G)$ színnel?**
Felveszünk egy $K_{\chi(G)}$-t G mellé, az egyes pontjai az egyes színeket jelölik. G első pontját hozzákötjük $K_{\chi(G)}$ minden egyes pontjához, kivéve egyhez. A „hozzákötetlen" pont fog megfelelni a választott színnek. Ezután megkérdezzük, hogy a gráf színezhető-e továbbra is $\chi(G)$ színnel. Ha igen, akkor megtaláltuk az adott pont színét, ha nem, akkor válaszunk újat. Ezt G összes pontjára megcsináljuk, mindegyikre legfeljebb az összes színt próbáljuk végig, így $n^2 \cdot poli$ lépésből kész vagyunk.

13. **Lássuk be, hogy a következő probléma NP-beli!**

$$\pi = \left\{ (s_1, s_2, \ldots, s_n, b) \middle| \forall i \ s_i, b \in \mathbb{Z}^+; \exists j_1, \ldots, j_k \ (1 \leq k \leq n) : \sum_{l=1}^{k} s_{j_l} = b \right\}$$

Tanú: $j_1, \ldots, j_k$ indexek. Tanú mérete: legfeljebb n, polinomiális. Ellenőrzés: kiszámoljuk $\sum_{l=1}^{k} s_{j_l}$ összeget, és megnézzük, hogy ez b-e; legfeljebb n összeadás, és egy hasonlítás, ami polinomiális.

14. **Bizonyítsuk be, hogy P-beli az olyan négy színnel színezhető G gráfokból álló nyelv, melyekre igaz, hogy G csúcsai kiszínezhetők a piros, zöld, sárga, kék színekkel úgy, hogy pontosan egy csúcs legyen piros és pontosan két csúcs legyen kék!**
A piros és kék színeket legfeljebb $n\binom{n-1}{2}$ féle módon oszthatjuk ki, ami polinomiális. Minden kiosztáshoz a maradék gráfot két színnel kell színezni, ami P-beli feladat. Így polinomszor kell egy polinom költségű algoritmust futtatni, ami polinom időben megy.