Algoritmuselmélet vizsgazárthelyi

A rendelkezésre álló munkaidő 100 perc. 2011. december 22.
Kérjük, minden résztvevő **nevét, NEPTUN kódját**, a dolgozat *minden* lapjának jobb felső sarkában *olvashatóan* és *helyesen* tüntesse fel. Ezen kívül a legfelső lapra írja rá **gyakorlatvezetője nevét** is (akihez a NEPTUN szerint jár).
Minden egyes feladat helyes megoldása 10 pontot ér. A dolgozatok értékelése: 0-31 pont: 1, 32-43 pont: 2, 44-55 pont: 3, 56-67 pont: 4, 68-80 pont: 5. A puszta (indoklás nélküli) eredményközlést nem értékeljük. A megindokolt részeredményért arányos pontszám jár. Az évvégi jegy kiszámításakor a (legalább elégséges) zh pontszámát vesszük figyelembe.
Írószeren és papírokon kívül semmilyen segédeszköz használata sem megengedett, így tilos az írott vagy nyomtatott jegyzet, a számoló- és számítógép ill. mobiltelefon használata, továbbá a dolgozatírás közbeni együttműködés.
Az eredményeket péntek estéig igyekszünk közzétenni a honlapon.
Megtekintés, szóbeli: 2012. január 3. kedd, 14:00-15:00, IB2.17.1.

1. Írd le a legrövidebb utak keresésére szolgáló Dijkstra-algoritmust. Mi az algoritmus alkalmazásának feltétele? (Az algoritmus helyességét nem kell bizonyítani.) Mennyi az algoritmus lépésszáma, ha a gráf a mátrixával van megadva és miért?

Egy lehetséges megoldás:
A Dijsktra-algoritmus egy súlyozott, de negatív súlyokat nem tartalmazó irányított (vagy irányítatlan) G gráf egy adott s pontjából meghatározza a legrövidebb utak hosszát a gráf többi pontjába. Az (x, y) él súlyát jelölje $C[x, y]$.

(1) KÉSZ := $\{s\}$
    for minden $v \in V$ csúcsra **do**
        $D[v] := C[s, v]$ ($*$ a $d(s, v)$ távolság első közelítése $*$)
(2) **for** $i := 1$ **to** $n - 1$ **do begin**
        Válasszunk olyan $x \in V \setminus$ KÉSZ csúcsot, melyre $D[x]$ minimális.
        Tegyük x-et a KÉSZ-be.
(3)   **for** minden $w \in V \setminus$ KÉSZ csúcsra **do**
            $D[w] := \min\{D[w], D[x] + C[x, w]\}$ ($*$ $d(s, w)$ új közelítése $*$)
    end

Lépésszám: Az (1) és (3) ciklus $O(n)$ lépés. A (2) ciklus $O(n)$ lépés, ez lefut $n - 1$-szer, tehát a lépésszám $O(n^2)$.

2. Definiáld a topologikus rendezés fogalmát. Milyen gráfoknak van topologikus rendezése? Ismertesd a topologikus rendezés megtalálására tanult algoritmust! (Indoklás nem szükséges.)

Egy lehetséges megoldás:
Legyen $G = (V, E)$ ($|V| = n$) egy irányított gráf. G egy *topologikus rendezése* a csúcsoknak egy olyan $v_1, \ldots, v_n$ sorrendje, melyben $x \to y \in E$ esetén x előbb van, mint y (azaz ha $x = v_i, y = v_j$, akkor $i < j$).

Egy irányított gráfnak akkor és csak akkor van topologikus rendezése, ha DAG.

Végezzük el a G DAG egy mélységi bejárását és írjuk ki G csúcsait a befejezési számaik szerint növekvő $w_1, \ldots, w_n$ sorrendben. A $w_n, w_{n-1}, \ldots, w_1$ sorrend a G DAG egy topologikus rendezése.

3. Definiáld a P és az NP problémaosztályt; mi az egymáshoz való viszonyuk? Válaszod indokold is meg.

Egy lehetséges megoldás:
P jelöli azoknak az eldöntési problémáknak a halmazát, amelyekhez van olyan $\mathcal{A}$ algoritmus és van olyan k pozitív konstans, hogy az algoritmus minden x bemenetre helyesen megválaszolja a kérdést, és az algoritmus lépésszáma *polinomiális*, azaz $O(|x|^k)$.
(Itt $|x|$ az x bemenet hosszát jelöli.)

NP azoknak az eldöntési problémáknak a halmaza, amelyekre van hatékony tanúsítvány. Azt mondjuk, hogy az X eldöntési problémához van *hatékony tanúsítvány*, ha van olyan $\mathcal{T}$ algoritmus, melynek a bemenete (x,t) párokból áll, ahol x az X probléma egy lehetséges bemenete és a következő feltételek teljesülnek: léteznek olyan c és k pozitív konstansok, hogy

- ha $x \in X$, akkor *van* olyan t, aminek hossza $|t| = O(|x|^c)$ és $\mathcal{T}(x,t) =$ igen,
- ha $x \notin X$, akkor *nincs* olyan t, aminek hossza $|t| = O(|x|^c)$ és $\mathcal{T}(x,t) =$ igen.
- A $\mathcal{T}$ algoritmus polinomiális, azaz a lépésszáma $O((|x| + |t|)^k)$.

Tudjuk, hogy $\mathrm{P} \subseteq \mathrm{NP}$, hiszen egy polimoniális algoritmus egy lefutása egyben hatékony tanúsítvány is. (Azt nem tudjuk, hogy $\mathrm{P} \neq \mathrm{NP}$ igaz-e.)

4. Adott egy n elemet tartalmazó és egy k elemet tartalmazó 2-3 fa. A két fában tárolt összes elemből $O(n+k)$ lépésben készíts egy rendezett tömböt.

Egy lehetséges megoldás:
A 2-3 fából lineáris időben kiolvasható a benne tárolt elemek listája növekvő sorrendben az inorder eljárással. (Ha a fában n elemet tárolunk, azaz n levele van, akkor belső csúcsa is legfeljebb ennyi lehet.) A kapott n és k elemű tömböt $n + k - 1$ összehasonlítással összefésülhetjük, így az összes elem egy rendezett tömbben lesz.

A lépészám $O(n) + O(k) + n + k - 1 = O(n + k)$.

5. Adott egy n pontú, e élű súlyozott, irányított gráf, a súlyok lehetnek negatívak is, de nincs negatív összsúlyú kör. Adott még a ponthalmaz két diszjunkt részhalmaza S és T. Adjunk $O(ne)$ lépésszámú algoritmust, ami meghatározza a legrövidebb olyan út összsúlyát, aminek kezdőpontja S-ben, végpontja pedig T-ben van!

Egy lehetséges megoldás:
A kapott G gráfból készítünk egy G' gráfot. Felveszünk egy új s pontot, amiből minden S-beli pontba mutasson egy 0 súlyú él. Felveszünk egy új t pontot is, minden T-beli pontból mutasson egy 0 súlyú él t-be. Keressük meg a Bellmann-Ford algoritmus segítségével a legrövidebb utat s-ből t-be. Ebből elhagyva s-et és t-t, a keresett utat kapjuk mert a G-beli és G'-beli utak kölcsönösen megfeleltethetők egymásnak és a megfelelő utak hosszai megegyeznek.

Lépésszám: G' pontjainak száma $n+2$, éleinek száma $\leq e+2n$, ezért G' elkészítése $O(n+e)$ lépés. A Bellmann-Ford lépésszáma pedig $O(n'e') = O((n+2)(e+2n)) = O(ne)$.

6. Adott a síkon n város: $v_1, v_2, \ldots, v_n$. Két város távolsága, $s_{i,j}$, a városok közötti euklideszi távolság. Két autó indul v_1-ből, minden városba el kell juttatniuk egy-egy ZH feladatsort (mindenhova egyformát, mindkét autóban van n példány). Mindkét autó által meglátogatott városok indexei csak növekvő sorozatot alkothatnak. Adjunk $O(n^3)$ futásidejű algoritmust, ami meghatározza a két autó által megtett út összegének minimumát!

Egy lehetséges megoldás:
Dinamikus programozással oldható meg a feladat.

Jelöljük $c(i,j)$-vel v_i és v_j távolságát. Ha $i < j$, akkor jelölje $D(i,j)$, hogy mi az a minimális távolság, amit az autók megtettek úgy, hogy az első j város mindegyikébe eljutott már a levél és az egyik autó v_i-ben van, a másik pedig v_j-ben.

Ha $i < j-1$, akkor v_j-ben ugyanaz az autó járt, mint v_{j-1}-ben. Ekkor a legrövidebb távot úgy kapjuk, hogy megnézzük a mi $D(i,j-1)$.

Ha $i = j-1$, akkor v_j-ben nem ugyanaz az autó járt, mint v_{j-1}-ben. Ekkor meg kell néznünk, hogy a legrövidebb út esetén melyik v_k városból ment autó v_j-be.

Nyilván $D(1,2) = c(1,2)$. Ha pedig $j \geq 3$, akkor

$$D(i,j) = \begin{cases} D(i,j-1) + c(j-1,j) & \text{ha } i < j-1 \\ \min_{1 \leq k < j-1}(D(k,j-1) + c(k,j)) & \text{ha } i = j-1. \end{cases}$$

Mivel egy $D(i,j)$ kiszámítása $O(n)$ lépés és a lehetséges i,j párok száma $O(n^2)$, ezért az összes $D(i,j)$ kiszámítása $O(n^3)$ lépés.

A keresett érték pedig $\min_{i<n}(D(i,n))$, ami ezután már $O(n)$ lépésben kiszámolható.

7. A kezdetben üres $M = 8$ méretű hashtáblába a $h(x) = 5x \pmod M$ hash-függvény és a $h'(x) = (x \pmod 3) + 1$ másodlagos hash-függvény segítségével az adott sorrendben rakd be a $3, 2, 4, 11, 5, 1$ elemeket. Ábrázold az egyes beszúrások menetét is!

Egy lehetséges megoldás:
Kettős hasshelésnél a próbasorozat: $h_i = -i \cdot h'(K)$. A $3, 2, 4, 1$ esetén a hashfüggvény által adott cella üres. 11 esetén a $7, 4, 1$ cellákat próbáljuk, az 5 esetén pedig az $1, 6$ cellákat.

0	1	2	3	4	5	6	7
	11	2		4			3

0	1	2	3	4	5	6	7
	11	2		4	1	5	3

8. Igazold, hogy a következő eldöntési probléma P-ben van, vagy azt, hogy NP-teljes:

Input: G gráf, melyre teljesül, hogy $e(G) \leq 3v(G)$

Kérdés: Igaz-e, hogy G kiszínezhető 3 színnel?

($e(G)$ a gráf éleinek, $v(G)$ a gráf pontjainak számát jelöli.)

Egy lehetséges megoldás:
Jelöljük a feladat problémáját X-el.

$X \in$ NP hiszen egy jó színezés hatékony tanúsítvány, mert röviden leírható és az ellenőrzése is gyors.

Belátjuk, hogy 3SZÍN $\prec X$. Legyen $f(G)$ egy olyan gráf, amit úgy kapunk, hogy G-hez hozzáveszünk $\lceil e/3 \rceil - v$ izolált pontot. Így $e' = e \leq 3v + 3(\lceil e/3 \rceil - v) = 3v'$ teljesül. Másrészt G nyilván akkor és csak akkor színezhető 3 színnel, ha $f(G)$ is. $f(G)$ polinom időben kiszámolható.

Tehát X egy NP-teljes probléma.