

# Algoritmusok elmélete

## 10. gyakorlat

2008. április 16.

- Éllistájukkal adottak a  $G_1$  és  $G_2$  irányított gráfok (zárójelben az élsúlyok).  
 $G_1$ : a:b(3),c(8); b:d(-7); c:d(5); d:e(2); e:a(-10);  
 $G_2$ : a:b(2),e(1),f(-2); b:c(8),e(-7); c:: d:c(- $\pi$ ); e:c(100),d(5); f:d(3), e(8);  
(a) Döntsük el mélységi bejárás segítségével, hogy ezek a gráfok DAG-ok-e!  
(b) Amelyik gráf DAG, abban adjunk meg egy topologikus sorrendet, határozzuk meg az a jelű csúcsból az összes többibe vezető legrövidebb utak hosszát és számítsuk ki a gráfban levő leghosszabb út hosszát is.
- Bizonyítsuk be, hogy minden hurokélmentes  $G = (V, E)$  irányított gráf felbontható két DAG-ra; pontosabban az élhalmazának van olyan  $E_1, E_2$  partíciója ( $E = E_1 \cup E_2$  és  $E_1 \cap E_2 = \emptyset$ ), hogy a  $G_1 = (V, E_1)$  és a  $G_2 = (V, E_2)$  gráfok DAG-ok!
- Cirkuszi akrobaták egymás vállára állva minél nagyobb tornyot szeretnének létrehozni (a toronyban minden szinten csak egy akrobata lesz). Esztétikai és gyakorlati szempontok miatt egy ember vállára csak olyan állhat, aki nála alacsonyabb és könnyebb is. A cirkuszban  $n$  akrobata van, adott mindegyikük magassága és súlya. Adjon algoritmust, amely  $O(n^2)$  lépésben megadja a lehetséges legtöbb emberből álló torony összeállítását.
- Legyen adott egy  $n \times n$  pixelből álló fekete-fehér kép. Szeretnénk a képen a bal felső saroktól a jobb alsó sarokig egy jobbra-lefele haladó határvonalat húzni úgy, hogy a vonaltól jobbra-lefele eső fekete, valamint a vonaltól balra-lefele eső fehér pixelek számának az összege a lehető legkisebb legyen. Oldjuk meg ezt a feladatot  $O(n^2)$  időben!
- Kutyasétáltatáskor egy parkban egy gazda egy rögzített, egyenes szakaszokból álló útvonalon halad, aminek töréspontjai  $t_1, \dots, t_n$ , a bejáratot jelölje  $t_0$ , a kijáratot  $t_{n+1}$ . A kutyája szabadon szaladgál, de a  $t_i$  pontokban találkozik a gazdájával. A  $t_i$  és  $t_{i+1}$  pontokban való találkozás között a kutya szeretne egy fát is meglátogatni (minden  $i = 0, 1, \dots, n$  esetén legfeljebb egyet-egyét). Legyenek adottak az  $s(t_i, t_{i+1})$  távolságok ( $0 \leq i \leq n$ ), valamint minden fának az összes  $t_i$  ponttól vett távolsága. Tegyük fel, hogy két találkozás között a kutya legfeljebb kétszer akkora távolságot tud megtenni, mint a gazda. Adjon algoritmust, ami segít a kutyának eldönteni, hogy mikor melyik fát látogassa meg ha a kutya célja, hogy minél több fánál járjon. Az algoritmus lépésszáma legyen  $O(n^2 f + n f^2)$ , ahol  $f$  a parkban levő fák számát jelöli.
- Egy előre rögzített útvonalon úgy indulunk el, hogy az autó  $L$  literes tankja tele van. Úticélunkhoz úgy akarunk elérni, hogy legalább egy fél tanknyi benzin maradjon az autóban. Tudjuk, hogy az utunkba eső  $n$  benzinkút közül melyikben mennyibe kerül a benzin, továbbá, hogy két szomszédos benzinkút között, valamint a kiindulóponttól az első benzinkútig, illetve az utolsó benzinkúttól a célunkig mennyi benzint fogyaszt az autó. Az egyszerűség kedvéért ha megállunk egy benzinkútnál, akkor mindig tele tankolunk. Adjon algoritmust, ami  $O(Ln^2)$  lépésben megmondja, hogy hol álljunk meg tankolni ha azt akarjuk, hogy utunk során a benzinköltség minimális legyen.
- Van  $n$  fájlunk, az  $i$ -edik fájl hosszát jelölje a  $h_i$ . Tegyük fel, hogy a  $h_i$  számok egészek. Mentéshez két egyformán  $L$  méretű lemez áll rendelkezésünkre ( $L$  pozitív egész szám). A cél, hogy minél nagyobb  $k$  számra az első  $k$  darab fájl mindegyikét mentsük ki a lemezekre. Fájlokat szétvágni nem szabad, minden fájl teljes egészében kerül az egyik vagy a másik lemezre. Adjon algoritmust, ami adott  $L$  és  $h_i$  számokhoz meghatározza, hogy melyik fájl melyik lemezre tegyük ahhoz, hogy  $k$  a lehető legnagyobb legyen. Az algoritmus lépésszáma legyen  $O(L^2)$ .