Bejárások

1. Adott a G irányítatlan gráf a következő éllistával: **a**: b, c; **b**: a, d; **c**: a, d; **d**: b, c, e, f; **e**: d, f, g; **f**: d, e, g, h; **g**: e, f, h; **h**: f, g. Készítsen G-ben egy a-ból kiinduló szélességi feszítőfát! Ez alapján mennyi az egyes csúcsok a-tól való távolsága?

2. Éllistával adott a súlyozott élű $G = (V, E)$ gráf. Tegyük fel, hogy minden él súlya az 1, 2, 3 számok közül való. Javasoljon $O(n + e)$ költségű algoritmust az $s \in V$ pontból az összes többi $v \in V$ pontokba vivő legrövidebb utak hosszának a meghatározására. (Itt n a G gráf csúcsainak, e pedig az éleinek a számát jelöli.)

3. Egy kezdő autóvezető a városban való közlekedése során szeretne gyakorlatának megfelelő útvonalat választani. Az úthálózat egy irányítatlan gráfként van megadva, a csúcsok a kereszteződések, az élek az utak, a csúcsoknál adott, hogy nehéz-e számára az a kereszteződés. (Az hogy nehéz, a kereszteződés tulajdonsága, nem azon múlik, merről érkezik oda és merre akar rajta áthaladni.) Írjon le egy algoritmust, amivel meg lehet határozni, hogy az autós az egyik adott csúcsnál levő otthonából mely csúcsokba tud autóval úgy eljutni, hogy útja során két nehéz csúcs soha nem jön közvetlenül egymás után. Az algoritmus lépésszáma éllistás megadás esetén legyen $O(n + e)$, ahol n a csúcsok és e az élek száma.

4. Adott a G irányított gráf a következő éllistával: **a**: b, c; **b**: -; **c**: d, e, f; **d**: f, g; **e**: b; **f**: -; **g**: h; **h**: d, b.

   Járja be G-t mélységi bejárással, az a-ból indulva, ennek során határozza meg a mélységi és befejezési számokat és osztályozza is a gráf éleit.

5. Éllistájukkal adottak az alábbi 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**: g(2), f(10); **b**: a(-2), g(1); **c**: -; **d**: -; **e**: c(5), d(6); **f**: e(7); **g**: f(1), e(8).

   (a) Mélységi bejárás segítségével döntse el, hogy ezek a gráfok DAG-ok-e! (b) Amelyik gráf DAG, abban adjon meg egy topologikus sorrendet, határozza meg az a jelű csúcsból a c-be vezető legrövidebb út hosszát és számítsa ki a gráfban levő leghosszabb út hosszát is!

6. A 6 pontú G gráf csúcsait jelölje x, y, z, u, v, w. A gráf egy mélységi bejárásánál a mélységi, ill. a befejezési számok a következők: x: 1, 6; y: 2, 4; z: 6, 5; u: 3, 3; v: 4, 1; w: 5, 2. Adja meg a bejáráshoz tartozó mélységi feszítőfa éleit. Rekonstruálható-e G gráf az előző számok ismeretében?

7. Éllistával adott a G gráf, ami egy n pontú és $e \geq n - 1$ élű DAG. Adjon $O(ne)$ lépésszámú algoritmust, ami minden i, j pontpárra meghatározza az i-ből j-be vezető utak számát.

8. Legyen G egy irányítatlan összefüggő gráf. Igaz-e, hogy (a) G minden f éléhez van G-nek olyan mélységi bejárása, amelyben f egy faél? (b) G minden f éléhez van G-nek olyan szélességi bejárása, amelyben f egy faél? (c) G minden F feszítőfájához van G-nek olyan mélységi bejárása, amelyben F minden éle faél? (d) G minden F feszítőfájához van G-nek olyan szélességi bejárása, amelyben F minden éle faél?

9. 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.

10. Van b darab borítékunk, az i-ediknek a hossza h_i, a magassága m_i. Az i-edik borítékba akkor tudjuk berakni a j-edik borítékot, ha $h_j < h_i$ és $m_j < m_i$ is teljesül (nem forgatjuk és nem is hajtogatjuk a borítékokat). Célunk, hogy minél hosszabb olyan láncot alakítsunk ki, hogy az i-edikben benne van a j-edik, abban a k-adik, stb. Hogyan lehet a maximális lánc hosszát $O(b^2)$ lépésben meghatározni?

11. Egy számítógéphálózatban n számítógép van. Minden olyan eseményt, hogy az i-edik gép üzenetet küld a j-ediknek (i, j, t) formában feljegyezünk, ahol a t egész szám az üzenet küldésének időpontját jelöli. Ugyanabban a t időpontban egy gép több gépnek is küldhet üzenetet. Ha a t időpontban az i-edik gép vírusos volt, akkor egy (i, j, t) üzenet hatására a j-edik gép megfertőződhet, ami azt jelenti, hogy a $t+1$ időponttól kezdve

már a j-edik gép is vírusos lehet. Legyen adott az (i, j, t) hármasoknak egy m hosszú listája, valamint x, y és $t_0 < t_1$ egész számok. Azt kell eldöntenünk, hogy ha az x-edik gép a t_0 időpontban vírusos volt, akkor lehet-e emiatt az y-adik gép a t_1 időpontban vírusos. Adjon algoritmust, ami ezt a kérdést $O)t_1 - t_0)n + m)$ lépés után megválaszolja.