Drumuri minime

+ Grafuri bipartite. Grafuri planare

Ca și în laboratorul trecut, fișierul `grafpond.in` **are următoarea structură: numărul de vârfuri n, numărul de muchii/arce m şi lista muchiilor/arcelor cu costul lor (o muchie fiind dată prin extremitățile sale și cost). Costul unei muchii este număr natural.**

grafpond.in
5 7
1 4 1
1 3 5
1 2 10
2 3 2
4 2 6
4 5 12
5 2 11

Justificaţi complexitatea+corectitudinea algoritmilor propuşi.

1. **Graf bipartit**. Se citesc din fişierul **`graf.in`** următoarele informaţi despre un graf neorientat (neponderat): numărul de vârfuri n, numărul de muchii m şi lista muchiilor (o muchie fiind dată prin extremităţile sale). Să se verifice dacă graful este sau nu bipartit. În caz afirmativ să se afișeze o bipartiție și să se studieze dacă graful este bipartit complet. În cazul în care graful nu este bipartit, să se afișeze **un ciclu elementar impar** al acestuia. **O(n+m)** **(1p)**

   Exemplu de aplicaţie: *O fabrică are la dispoziție două mașini pentru a transporta n substanțe numerotate 1, 2, ..., n. Pentru siguranța transportului în caz de eventuale accidente, este întocmită o listă de perechi (i, j) de substanțe care interacționează. Să se determine dacă există o modalitate de a încărca aceste substanțe în cele două mașini pentru a fi transportate, astfel încât în fiecare dintre cele două mașini să nu se găsească substanțe care interacționează (altfel spus, două substanțe care interacționează nu sunt transportate cu aceeași mașină). Dacă există o astfel de modalitate, să se afișeze substanțele care vor fi încărcate în prima mașină și substanțele care vor fi încărcate în cel de a doua mașină.*

2. **Drum critic (Critical Path Method).** Se citesc din fișierul `activitati.in` următoarele informații despre activitățile care trebuie să se desfășoare în cadrul unui proiect:
   - n – numărul de activități (activitățile sunt numerotate 1,…, n)
   - d_1, d_2, …., d_n durata fiecărei activități
   - m – număr natural
   - m perechi (i, j) cu semnificația: activitatea i trebuie să se încheie înainte să înceapă j

   Activitățile se pot desfășura și în paralel.

   a) Să se determine timpul minim de finalizare a proiectului, știind că acesta începe la ora 0 (echivalent – să se determine durata proiectului) și o succesiune (critică) de activități care determină durata proiectului (un drum critic – v. curs) **O(m + n).** **(2p)**

b) Să se afişeze pentru fiecare activitate un interval posibil de desfăşurare (!ştiind că activitățile se pot desfăşura în paralel) **O(m + n).** **(1p)**

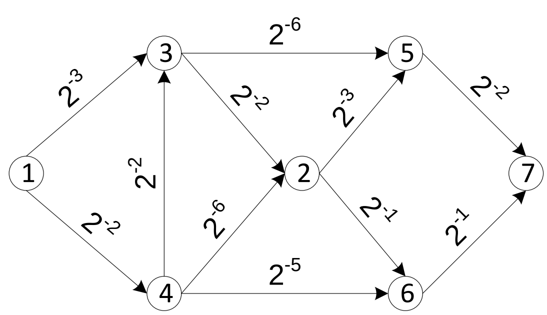
Robert Sedgewick and Kevin Wayne, Algorithms, 4th Edition, Addison-Wesley, 2011.

activitati.in	iesire
6 **7 4 30 12 2 5** **6** **1 2** **2 3** **3 6** **4 3** **2 6** **3 5**	Timp minim 47 Activitati critice: 4 3 6 1: 0 7 2: 7 11 3: 12 42 4: 0 12 5: 42 44 6: 42 47

Algoritmul lui Dijkstra – două probleme la alegere între 3,4,5.

3. Se citesc din fişierul `grafpond.in` informații despre un graf **neorientat** ponderat și de la tastatură un număr k, o listă de k puncte de control ale grafului şi un vârf s. Determinați cel mai apropiat punct de control de vârful s şi afişaţi un lanț minim până la acesta, folosind algoritmul lui **Dijkstra** (problema B.1. din laboratorul 1 pentru cazul ponderat) - **O(m log(n)).** **(3p)**

4. Pentru fiecare arc al unei reţele de comunicaţie acestui graf se cunoaşte o pondere pozitivă subunitară reprezentând probabilitatea ca legătura corespunzătoare să nu se defecteze (de forma $1/2^p = 2^{-p}$). Aceste probabilităţi sunt independente, deci **siguranţa unui drum** este egală cu produsul probabilităţilor asociate arcelor care îl compun. Arătaţi că problema determinării unui drum de siguranţă maximă de la un vârf de start s la un vârf destinaţie t (accesibil din s) se poate reduce la o problemă de determinare a unui drum minim între s şi t (pentru un graf cu ponderile modificate). Pornind de la acest fapt, implementaţi un algoritm bazat pe algoritmul lui Dijkstra pentru determinarea unui drum de siguranţă maximă între două vârfuri s şi t citite de la tastatură pentru o reţea orientată dată în fişierul retea.in prin următoarele informaţii:
   - n, m – numărul de vârfuri, respectiv arce
   - m linii conţinând triplete de numere naturale i j p cu semnificaţia: (i,j) este arc în reţea cu probabilitatea să nu se defecteze egală cu 2^{-p} **O(m log(n)).** **(2p)**



5. http://www.infoarena.ro/problema/catun **O(m log(n)).** **(3p)**

SUPLIMENTAR (*Se punctează doar dacă s-a obținut un minim de 8,5 puncte din cele 4 probleme precedente - 2 obligatorii si 2 la alegere)*

6. **Graf planar - 6-colorare** Se citesc din fişierul `graf.in` următoarele informaţi despre un graf neorientat planar: numărul de vârfuri n, numărul de muchii m şi lista muchiilor (o muchie fiind dată prin extremităţile sale). Să se afişeze o 6-colorare proprie a acestuia. **O(n+m)** (**1.5p**)