Drumuri minime

Ca şi în laboratorul trecut, fişierul `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 extremităţile sale şi 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. **Drum critic (Critical Path Method).** Se citesc din fişierul `activitati.in` următoarele informații despre activitățile care trebuie să se desfăşoare 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) şi o succesiune (critică) de activități care determină durata proiectului (un drum critic – v. curs) **O(m + n).**

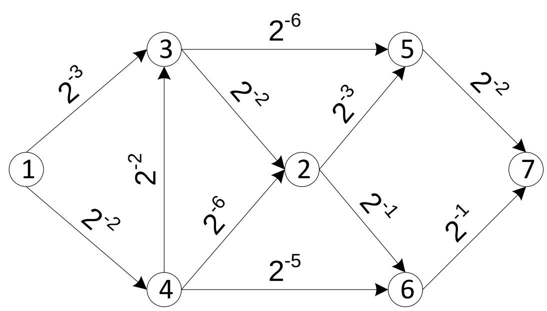
   b) Să se afişeze pentru fiecare activitate un interval posibil de desfăşurare (!știind că activitățile se pot desfășura în paralel) **O(m + n).**

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

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

2. Se citesc din fişierul `grafpond.in` informaţii despre un graf **neorientat** ponderat şi 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)).**

3. 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)).**



4. **Drumuri minime din surse multiple** http://www.infoarena.ro/problema/catun **O(m log(n)).**

5. **Centru, rază, diametru**. Se citesc din fișierul `grafpond.in` informații despre un graf **neorientat** ponderat G. Noțiunile de excentricitate a unui vârf, centru și diametru pentru un graf ponderat se definesc similar cu cele din cazul neponderat (dar cu distanța definită în funcție de costul muchiilor). Să se determine, folosind algoritmul **Floyd-Warhsall**, raza, diametrul, centrul grafului și un lanț diametral (un lanț minim P între două vârfuri u și v cu ponderea w(P)=d(u,v)=diam(G)).