

Subiectul 1 (3 puncte)

Se dă un arbore ponderat T cu $n > 3$ vârfuri și un vârf s .

Informațiile despre arbore se citesc din fișierul **graf.in** cu structura:

- pe prima linie este n
- pe următoarele n linii sunt listele de adiacență ale lui G ; linia i începe cu gradul vârfului i și apoi conține vecinii vârfului i , fiecare vecin fiind urmat de costul muchiei:

<grad> <vecin1> <cost_muchie_de_la_i_la_vecin1> <vecin2> <cost_muchie_de_la_i_la_vecin2>
etc

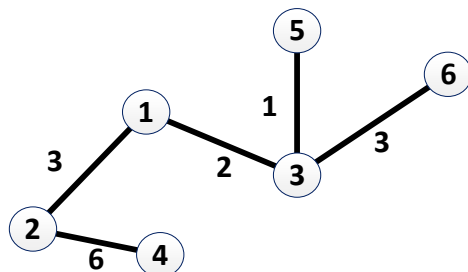
- pe ultima linie este vârful s

(vârfurile sunt numerotate de la 1)

Pentru un lanț P în T definim capacitatea lanțului P ca fiind capacitatea minimă a unei muchii din P .

- Să se afișeze pentru fiecare vârf v capacitatea unicului lanț elementar de la s la v (sub forma v : capacitate lanț) **Complexitate $O(n)$**
- Să se afișeze care este capacitatea minimă a unui lanț cu o extremitate în s și să se afișeze un astfel de lanț

<i>graf.in</i>	<i>iesire pe ecran</i>
6 2 2 3 3 2 2 1 3 4 6 3 1 2 5 1 6 3 1 2 6 1 3 1 1 3 3 1	1: 0 2: 3 3: 2 4: 3 5: 1 6: 2 Capacitatea minima este 1 pentru lantul 1 3 5
Explicații: informațiile de pe linia 2 2 3 3 2 corespunzătoare vârfului 1 se citesc astfel: varful 1 are gradul 2; primul său vecin este 2 și costul muchiei de la el la 2 este 3; al doilea sau vecin este 3, iar costul muchiei de la el la 3 este 2	



Subiectul 2 (3 puncte)

Se citesc informații despre un graf **neorientat** ponderat conex G din fișierul `graf.in`.

Fișierul are următoarea structură:

- pe prima linie sunt două numere reprezentând numărul de vârfuri n ($n > 4$) și numărul de muchii m ale grafului, $m > n$
- pe următoarele m linii sunt câte 3 numere pozitive reprezentând extremitatea inițială, extremitatea finală și costul unei muchii din graf

a) Să se afișeze costul unui arbore parțial de cost minim în G. **Complexitate $O(m \log(n))$.**

b) Se citesc de la tastatură două muchii **noi** date tot prin extremitatea inițială, extremitatea finală și cost. Știind că **doar una** dintre aceste muchii se va adăuga la graful G, decideți pe care o adăugați astfel încât noul graf să aibă un arbore parțial de cost minim cu cost cât mai mic și afișați muchiile unui arbore parțial de cost minim în acest graf. **Complexitate $O(n)$**

Exemplu

<code>graf.in</code>	Iesire pe ecran (nu conteaza ordinea în care sunt afisate muchiile)
5 5 1 2 1 1 4 2 2 3 4 3 4 8 4 5 6 Intrare de la tastatura 3 5 5 1 3 5	a) 13 b) adaugam 3 5 1 2 1 4 2 3 3 5



Subiectul 3 (3 puncte)

Propuneți un algoritm bazat pe algoritmul Ford-Fulkerson / Edmonds Karp pentru rezolvarea următoarei probleme.

Fișierul graf.in conține următoarele informații despre un graf **bipartit** conex cu $V_1 = \{1, \dots, p\}$ și $V_2 = \{p+1, \dots, n\}$:

- pe prima linie sunt 2 numere naturale n și m reprezentând numărul de vârfuri și numărul de muchii
- pe a doua linie este p
- pe următoarele m linii sunt perechi de numere x y (separate prin spațiu) reprezentând extremitățile unei muchii, $x \in V_1$ și $y \in V_2$.

Scrieți un program care citește datele despre graful G din fișierul graf.in și afișează:

a) Un cuplaj de cardinal k în G , cu k citit de la tastatură. Dacă nu există un astfel de cuplaj se va afișa mesajul “nu exista” **Complexitate $O(km)$**

b) Muchiile unui 2-factor în G , dacă există (2-factor = graf parțial în care toate vârfurile au gradul 2) **Complexitate $O(nm)$**

graf.in	Iesire pe ecran (soluția nu este unică)
8 10	a)
4	pentru $k=2$
1 5	1 5
1 6	2 6
1 7	b) un 2-factor:
2 5	1 5
2 6	1 6
3 5	2 5
3 7	2 6
3 8	3 7
4 7	3 8
4 8	4 7
	4 8

