

Subiectul 1 (3 puncte)

Se dă un graf neorientat ponderat G cu $n > 3$ vârfuri, m muchii și un vârf s .

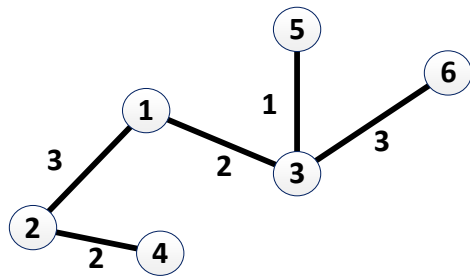
Informațiile despre graf se citesc din fișierul **graf.in** cu structura:

- pe prima linie sunt n și m
- pe următoarele m linii sunt câte 2 numere naturale reprezentând extremitățile unei muchii
- pe ultima linie este un vârf sursă s .

Pentru un lanț P în G definim ponderea lanțului P ca fiind produsul ponderilor muchiilor care îl compun.

Dacă G este arbore, să se afișeze pentru fiecare vârf v ponderea unicului lanț elementar de la s la v (sub forma v : pondere lanț de la s la v), altfel să se afișeze un arbore parțial al componente care conține s . **Complexitate $O(n+m)$**

<i>graf.in</i>	<i>iesire pe ecran</i>
6 5	Este arbore
1 2 3	1: 0
1 3 2	2: 3
2 4 2	3: 2
3 5 1	4: 6
3 6 3	5: 2
1	6: 6



Subiectul 2 (3 puncte)

Se citesc informații despre un graf **orientat** ponderat **fără circuite** 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 arce m ale grafului, $m > n$
- pe următoarele m linii sunt câte 3 numere întregi reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf (**costul poate fi și negativ**)
- pe următoarea linie (a $(m+2)$ -a linie) din fișier este un număr natural k ($0 < k < n$) reprezentând numărul de vârfuri sursă; vârfurile sursă din G vor fi $1, 2, \dots, k$
- pe ultima linie a fișierului sunt două vârfuri t_1 și t_2 , reprezentând vârfurile destinație ale grafului (distincte de vârfurile sursă din G).

Notăm cu $S = \{1, \dots, k\}$ mulțimea vârfurilor sursă din G și cu $T = \{t_1, t_2\}$ mulțimea vârfurilor destinație din G. Spunem că un vârf y este accesibil din x în G dacă există un drum de la x la y . Presupunem că există cel puțin un vârf destinație care este accesibil dintr-un vârf sursă.

- Să se verifice dacă graful dat este fără circuite și să se afișeze un mesaj corespunzător.
- Să se determine distanța între cele două mulțimi S și T:

$$d(S, T) = \min \{d(x, y) \mid x \in S, y \in T\}$$

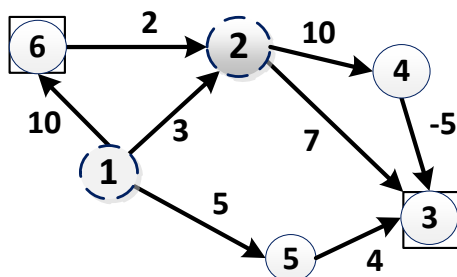
Să se determine în plus și o pereche de vârfuri (s, t) cu $s \in S$ și $t \in T$ cu

$$d(s, t) = d(S, T) = \min \{d(x, y) \mid x \in S, y \in T\}$$

și să se afișeze (pe ecran) un drum minim de la s la t . **Complexitate $O(m)$**

Exemplu

graf.in	Iesire pe ecran
<pre> 6 8 1 2 3 1 6 10 6 2 2 2 4 10 4 3 -5 5 3 4 1 5 5 2 3 7 2 3 6 </pre>	<pre> distanța între multimi = 5 s=2 t=3 drum minim 2 4 3 </pre>



Explicații

$k=2 \Rightarrow S = \{1, 2\}$

$T = \{3, 6\}$

$d(1,3)=8, d(2,3)=5$

$d(1,6)=10, d(2,6)=\infty$

Cea mai mică este $d(2,3)=5$

Un drum minim de la 2 la 3 este 2 4 3

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

