## **Subjectul 1 (3 puncte)**

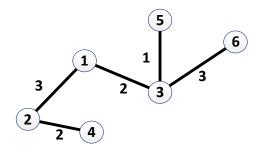
Se dă un graf neorientat ponderat G cu n>3 vârfuri, m muchii și un vârf s. Informațiile despre graf se citesc din fișierul *graf.in* cu structura:

- 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 componentei care conține s. **Complexitate O(n+m)** 

graf.in	lesire pe ecran	
65	Este arbore	
123	1:0	
132	2: 3	
2 4 2	3: 2	
351	4: 6	
363	5: 2	
1	6: 6	



# Subjectul 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 pozitive reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf
- pe următoarea linie (a (m+2)-a linie) din fișier sunt un număr natural k (0<k<n) și un șir de k vârfuri reprezentând vârfurile sursă ale grafului s<sub>1</sub>,...,s<sub>k</sub>
- pe ultima linie a fișierului sunt două vârfuri t<sub>1</sub> și t<sub>2</sub>, reprezentând vârfurile destinație ale grafului (distincte de vârfurile sursă de pe linia anterioară).

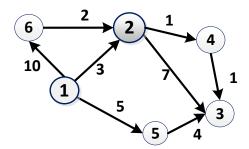
Notăm cu  $S = \{s_1,...,s_k\}$  mulțimea vârfurilor sursă din G și cu  $T = \{t_1, t_2\}$  mulțimea vârfurilor destinație din G. Spunem că un vârf y este accesibil din x în G dacă există un drum de la x la y.

- a) Să se verifice dacă graful dat este fără circuite și să se afișeze un mesaj corespunzător.
- b) Să se determine un drum de cost maxim care începe cu un vârf din S și se termină cu un vârf din T. Dacă nu există un astfel de drum se va afișa un mesaj corespunzător: niciun vârf destinație nu este accesibil dintr-un vârf sursă.

### **Complexitate O(m)**

#### Exemplu

2. Action 1910		
Iesire pe ecran		
drum maxim 1 6 2		



$$S = \{1, 3\}$$
  
 $T = \{2, 5\}$ 

## Subjectul 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  $V_1=\{1,...,p\}$  și  $V_2=\{p+1,...,n\}$ :

- 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	lesire pe ecran (solutia nu este unica)
8 10	a)
4	pentru k=2
15	15
16	2 6
17	b) un 2-factor:
25	15
2 6	16
3 5	2 5
3 7	2 6
3 8	3 7
4 7	3 8
48	4 7
	48

