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

Se dă un graf neorientat cu n>3 vârfuri și m muchii care nu este bipartit.

- a) Să se determine un ciclu elementar impar în graf (cu număr impar de muchii). Se vor afișa muchiile unui astfel de ciclu. **Complexitate O(n+m)**
- b) Să se determine dacă în grad mai există în graf un alt ciclu în afară de cel afișat la punctul
- a) (nu neapărat impar) și, în caz afirmativ, să se afișeze un astfel de ciclu (diferit de cel de la
- a)); altfel se va afișa un mesaj corespunzător. Complexitate O(n+m)

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

graf.in	lesire pe ecran
5 6	a)
12	2 4
23	45
2 4	5 2
4 5	b)
25	12
15	2 4
	4 5
	5 1

# 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).

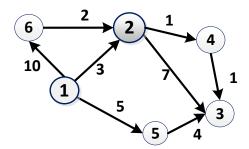
Notăm cu  $S = \{s_1, ..., s_k\}$  mulțimea vârfurilor sursă din G și cu  $T = \{t_1, t_2\}$  mulțimea vârfurilor destinație din G. Spunem că un vârf y este accesibil din x în G dacă există un drum de la x la y.

- 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

Exemple	
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.

Într-un restaurant sunt n mese numerotate 1,...,n sunt și m ospătari numerotați 1,..., m (m≥n). Proprietarul restaurantului urmează să aibă un eveniment în restaurant și dorește să repartizeze fiecărui ospătar mesele de care trebuie să se ocupe. El întreabă pe fiecare ospătar la câte mese ar vrea să servească maxim și notează cu o₁,...,o<sub>m</sub> răspunsurile acestora.

Proprietarul ar vrea ca la fiecare masa i să **fie exact k**i **ospătari** și ar vrea ca numărul de mese la care repartizează un ospătar i **să nu depășească** opțiunea acestuia o<sub>i</sub>.

Scrieți un program care, dacă este posibilă o distribuție a ospătarilor la mese care să respecte dorințele proprietarului, să afișeze o astfel de distribuție sub forma prezentată în exemplul de mai jos. Altfel se va afișa mesajul "imposibil"

Datele despre restaurant și opțiunile ospătarilor se vor citi dintr-un fișier cu următoarea structură:

- pe prima linie sunt numerele naturale n, m
- pe următoarea linie n numere naturale k<sub>1</sub>, ..., k<sub>n</sub> reprezentând câți ospătari trebuie să fie la fiecare masă
- pe următoarea linie m numere naturale o<sub>1</sub>...o<sub>m</sub> reprezentând opțiunile ospătarilor.

### Complexitate $O(n^2m^2)$

restaurant.in	lesire pe ecran (solutia nu este unica)
3 3	Masa 1: ospătari 1
121	Masa 2: ospătari 2 3
122	Masa 3: ospătari 3
restaurant.in	lesire pe ecran
4 4	Imposibil
1144	•
1144	