

## Subiectul 1 (3 puncte)

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

a) Adăugați la  $G$  un număr minim de muchii astfel încât să devină conex. Construiți în memorie și afișați pe ecran listele de adiacență ale grafului astfel obținut. **Complexitate  $O(n+m)$**

b) Determinați excentricitatea  $\text{ecc}(s)$  a vârfului  $s$  în noul graf  $G_1$  obținut la a):

$$\text{ecc}(s) = \max(d(s,v) \mid v \text{ vârf în } G_1)$$

unde  $d(s,v)$  este distanța de la  $s$  la  $v$ .

**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
- pe ultima linie este vârful  $s$

<i>graf.in</i>	<i>iesire pe ecran (soluția nu este unică)</i>
6 4	a)
1 3	1 2
1 5	2 6
3 5	b)
2 4	3
6	Explicații: $d(6,1)=2$ , $d(6,2)=1$ , $d(6,3)=3$ , $d(6,4)=2$ , $d(6,5)=3$ , deci $\text{ecc}(6)=3$

## 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 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  $s_1, \dots, s_k$
- pe ultima linie a fișierului sunt două vârfuri  $t_1$  și  $t_2$ , reprezentând vârfurile destinație ale grafului (distincte de vârfurile sursă de pe linia anterioară).

Notăm cu  $S = \{s_1, \dots, s_k\}$  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  $y$ .

- Să se verifice dacă graful dat este fără circuite și să se afișeze un mesaj corespunzător.
- 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

graf.in	Iesire pe ecran
6 8 1 2 3 1 6 10 6 2 2 2 4 1 4 3 1 5 3 4 1 5 5 2 3 7 2 1 3 2 5	drum maxim 1 6 2



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

### Subiectul 3 (3 puncte)

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

Într-un restaurant sunt  $n$  mese numerotate  $1, \dots, n$  sunt și  $m$  ospătari numerotați  $1, \dots, m$ .

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 ce mese ar vrea să servească, altfel că pentru fiecare ospătar  $j$  știe lista meselor pe care le preferă.

Proprietarul ar vrea ca la fiecare masă să fie exact  $k$  ospătari și un ospătar să servească la cel mult  $p$  mese și ar vrea să respecte și preferințele ospătarilor legate de mese.

Scrieți un program care, dacă este posibilă o distribuție a ospătarilor la mese care să respecte dorințele proprietarului, cu **cel mult** o excepție și anume o masă la care să fie doar  $k-1$  ospătari, să afișeze o astfel de distribuție sub forma prezentată în exemplul de mai jos (perechi masa ospătar). 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  $n, m, k, p$
- pe următoarele  $m$  linii sunt numere naturale, pe linia a  $j$ -a din cele  $m$  fiind numărul de mese la care vrea să servească ospătarul  $j$  și care sunt indicii acestor mese

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

restaurant.in	lesire pe ecran (solutia nu este unica)
4 7 2 1 4 1 2 3 4 1 1 2 2 4 2 2 3 2 1 4 3 1 3 4 1 1	1 2 1 7 2 3 2 4 3 1 3 6 4 5
Explicații: Ospătarul 1 prefera 4 mese: 1,2,3,4, ospătarul 2 preferă masa 1, ospătarul 3 preferă 2 mese: 2 și 4 etc	(primul indice reprezintă masa, iar al doilea ospătarul asociat)

