

Subiectul 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 graf 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

<i>graf.in</i>	<i>iesire pe ecran</i>
5 6	a)
1 2	2 4
2 3	4 5
2 4	5 2
4 5	b)
2 5	1 2
1 5	2 4
	4 5
	5 1

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.

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

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 o_1, \dots, o_m 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_i .

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_1, \dots, k_n reprezentând câți ospătari trebuie să fie la fiecare masă
- pe următoarea linie m numere naturale $o_1 \dots o_m$ reprezentând opțiunile ospătarilor.

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

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