

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 vârfurile 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 **neorientat** ponderat conex G din fișierul `graf.in`.

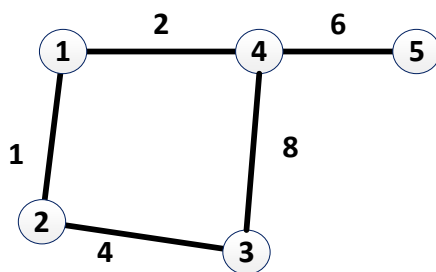
Fișierul are următoarea structură:

- 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
- pe ultima linie sunt un număr natural k ($0 < k < n$) și un șir de k vârfuri reprezentând vârfurile sursă ale grafului s_1, \dots, s_k

Să se afișeze muchiile unui graf parțial al lui G în care fiecare vârf v este conectat prin cel puțin un lanț de o sursă. Dacă există mai multe astfel de grafuri parțiale, să se determine unul cu costul total minim. Se vor afișa muchiile acestui graf parțial. De asemenea, să se afișeze și care este vârful sursă cel mai important **în acest graf parțial**, importanța unui vârf sursă fiind dată de numărul de vârfuri accesibile din acea sursă (care sunt conectate la acea sursă prin lanț). **Complexitate $O(m \log(n))$.**

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 2 3 4 3 4 8 4 5 6 2 1 2 | 1 4 4 5 2 3 1 |



Două surse: 1 și 2

În graful parțial cu muchiile

1 4

4 5

2 3

din sursa 1 sunt accesibile încă două vârfuri (4 și 5), iar din

2 doar unul (3), deci 1 este mai important

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 |