Subiectul 1

Se dă un graf neorientat conex cu n>3 vârfuri și m>n muchii.
Informațiile despre graf se citesc din fișierul graf.in cu următoarea structură:

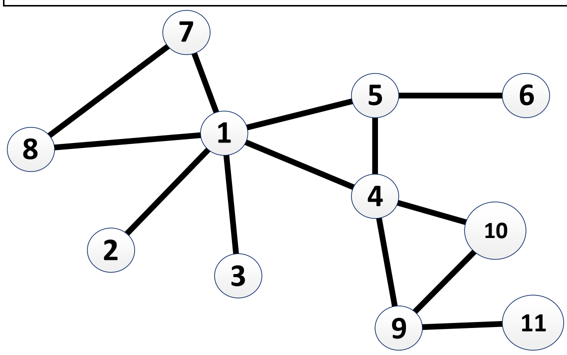
- pe prima linie sunt n și m
- pe următoarele m linii sunt câte 2 numere naturale reprezentând extremitățile unei muchii

Se citește de la tastatură un vârf v.

a) Să se afișeze muchiile incidente în v care **nu sunt critice**, dacă există (altfel se va afișa mesajul "nu exista"). O(**m**)

b) Să se afișeze muchiile unui arbore parțial T al lui G în care vârful v are gradul cu 1 mai mic decât îl are în G: $d_T(v) = d_G(v) - 1$, dacă un astfel de arbore există.

graf.in	Iesire pe ecran dacă se citește pentru v valoarea 1 (nu contează ordinea în care se afișează muchiile; soluția la b) nu este unică)
11 13 1 2 1 3 1 4 1 5 4 5 5 6 1 7 7 8 1 8 4 9 9 10 10 4 9 11	muchii care nu sunt critice: 1 4 1 5 1 7 1 8 Arbore: 1 2 1 3 1 4 1 5 5 6 1 7 7 8 4 9 4 10 9 11



Subiectul 2

Se citesc informații despre un graf **orientat fără circuite** G din fișierul `graf.in`.
Fișierul are următoarea structură:

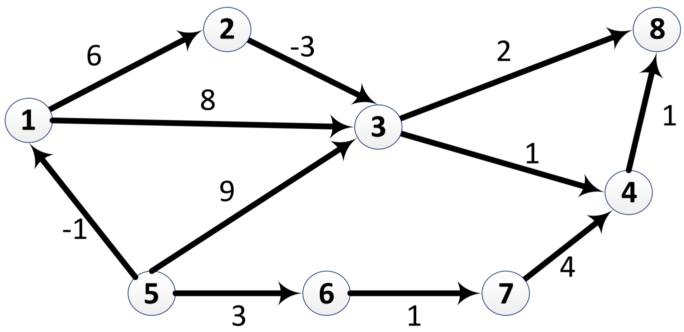
- pe prima linie sunt două numere reprezentând numărul de vârfuri n (n>4) și numărul de arce m ale grafului
- pe următoarele m linii sunt câte 3 numere întregi reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf (costul unui arc poate fi și **negativ**).
- pe ultima linie este un nod sursă s.

Spunem că un vârf y este accesibil din x în G dacă există un drum de la x la y.

a) Să se determine centralitatea vârfului s: cc(s) = suma distanțelor de la s la vârfurile t accesibile din s (=sum{d(s,t) | t este accesibil din s}). **Complexitate O(n+m)**

b) Să se afișeze un drum de cost maxim în G. **Complexitate O(n+m)**

graf.in	Iesire pe ecran (nu este unică)
8 11 1 2 6 2 3 -3 1 3 8 3 8 2 3 4 1 4 8 1 5 1 -1 5 3 9 5 6 3 6 7 1 7 4 4 1	18 5 3 8



Explicații:

d(1,2) = 6, d(1,3)=3, d(1,4) = 4, d(1,8) = 5 => cc(1) =6+3+4+5= 18

Drumul 5,3,8 are cel mai mare cost dintre toate drumurile din graf

Subiectul 3

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

La o conferință cu m sesiuni, numerotate 1,...,m sunt n invitați, numerotați 1,...,n. Un invitat poate participa la mai multe sesiuni. O sesiune poate fi coordonată de cel mult doi dintre invitații care participă la ea. Să se desemneze, dacă este posibil, ce sesiuni va coordona fiecare invitat, astfel încât orice invitat să coordoneze cel puțin o sesiune (!pot rămâne și sesiuni fără coordonatori, ce se cere este să nu rămână invitați care nu coordonează nicio sesiune). Datele despre invitații de la fiecare sesiune se vor citi dintr-un fișier cu următoarea structură:

- pe prima linie sunt numerele naturale n și m
- pe următoarele linii sunt perechi de numere naturale i j cu $i \in \{1,\ldots, n\}$ și $j \in \{1,\ldots, m\}$ cu semnificația: invitatul i participă la sesiunea j

Dacă se pot desemna coordonatori astfel încât orice invitat să coordoneze cel puțin o sesiune, rezultatul se va afișa sub forma prezentată în exemplul de mai jos. Altfel se va afișa mesajul "nu este posibil" și numărul minim de invitați care nu pot fi desemnați coordonatori.

Complexitate $O(n^2m)$ (pot fi cel mult mn perechi i j cu semnificația: invitatul i participă la sesiunea j)

Observație: Putem modela problema cu un graf bipartit invitați-sesiuni. O sesiune poate fi coordonată de cel mult un invitat, atunci problema se reduce la a determina un cuplaj de cardinal maxim în graful bipartit invitați-sesiuni și a verifica dacă orice vârf de tip invitat este saturat. Se acorda 1p dacă se rezolva problema pe cazul în care o sesiune poate fi coordonată de cel mult un invitat.

sesiuni.in	Iesire pe ecran (solutia nu este unica)
7 4 1 1 2 1 5 1 6 1 7 1 1 2 3 2 4 2 3 3 4 3 6 3 1 4 3 4 5 4	1 4 2 1 3 2 4 2 5 4 6 3 7 1 Dacă invitatul 5 nu ar mai participa la sesiunea 4, problema nu are soluție (cel puțin un invitat nu poate fi desemnat coordonator)