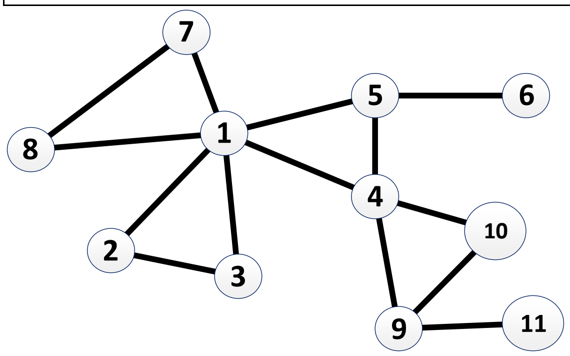
Subiectul 1

Se dă un graf neorientat conex cu n>3 vârfuri şi m>n muchii. Să se afişeze punctele critice în care **nu** sunt incidente muchii critice. Pentru fiecare astfel de punct se va afişa numărul de componente biconexe care îl conțin, fără a memora componentele biconexe ale grafului şi fără a memora muchiile critice. O(**m**)

Informațiile despre graf se citesc din fișierul graf.in cu structura:

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

graf.in	Iesire pe ecran (nu neaparat in aceasta ordine)
11 14 1 2 1 3 2 3 1 4 1 5 4 5 5 6 1 7 7 8 1 8 4 9 9 10 10 4 9 11	Puncte critice cerute: 1 – continut in 3 componente biconexe 4 - continut in 2 componente biconexe

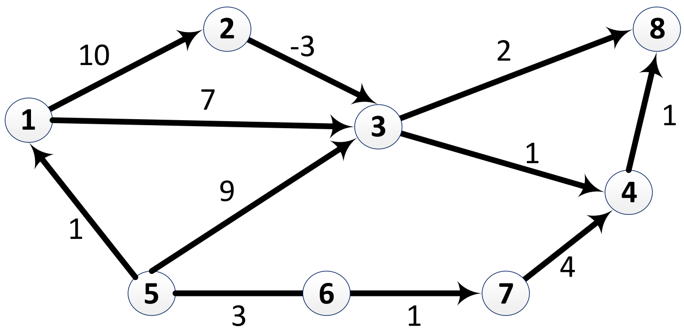


Subiectul 2

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, `m>=n`
- 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 sunt două noduri sursa s_1 și s_2
  a) Să se determine dacă există un vârf din graf v egal depărtat de s_1 și s_2: $d(s_1,v)=d(s_2,v)$. Dacă există mai multe astfel de vârfuri se va afișa cel mai apropiat de cele două surse (cel cu $d(s_1,v)$ cea mai mică). **Complexitate O(m)**
  b) Pentru vârful v determinat la a) (dacă există) să se determine dacă există mai multe drumuri minime de la s_1 la v. Daca exista doar unul, se va afișa acest drum, dacă nu se vor afișa două dintre drumurile minime de la s la v_1 **Complexitate O(m)**

graf.in	Iesire pe ecran
8 11 1 2 10 2 3 -3 1 3 7 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 5	a) v=4 b) 1 2 3 4 1 3 4 Explicații: d(1,4) = d(5,4) = 8



Subiectul 3

Se dau n fabrici de monitoare numerotate 1...n și m depozite numerotate n+1,.., n+m. Pentru fiecare fabrica i se cunoaște c(i) = câte monitoare au fost produse la momentul curent, iar pentru fiecare depozit j se cunoaște c(j) = numărul de monitoare pe care le poate depozita la momentul curent. Fiecare fabrică are contracte cu anumite depozite. În contractul dintre fabrica i și depozitul j este trecută cantitatea maximă de monitoare care poate fi trimisă spre depozitare de la fabrica i la depozitul j, notată w(i,j). Datele se vor citi din fișierul fabrici.in cu următoarea structură:

- pe prima linie sunt numerele naturale n și m
- pe a doua linie este un șir de n numere naturale reprezentând cantitatea de monitoare existente în fiecare dintre cele n fabrici
- pe a treia linie este un șir de m numere naturale reprezentând numărul de monitoare pe care le poate depozita fiecare dintre cele m depozite
- pe a patra linie este un număr k reprezentând numărul de contracte dintre fabrici și depozite
- pe următoarele k linii sunt triplete de numere naturale i j w (separate prin spatiu) cu semnificația: de la fabrica i la depozitul j se pot trimite maxim w monitoare.

Să se determine, dacă există, o modalitate de a depozita toate monitoarele existente în fabrici la momentul curent în depozite respectând condițiile din contracte și capacitatea de depozitare a fiecărui depozit. **Complexitate O((n+m)k^2)**

Rezultatul se va afișa sub forma prezentată în exemplul de mai jos.

Observație: Putem modela problema cu un graf bipartit fabrici-depozite (cu vârfuri corespunzătoare fabricilor și depozitelor și muchii reprezentând existența unui contract între fabrică și depozit). Dacă c(i) = 1 pentru fiecare fabrică i, c(j)=1 pentru fiecare depozit și w(i, j)=1 pentru orice contract, atunci problema se reduce la a determina un cuplaj de cardinal maxim în graful bipartit fabrici-depozite și a verifica dacă orice vârf fabrică este saturat.

Se acorda 1p daca se rezolva doar problema pentru c(i) = 1 pentru fiecare fabrică i, c(j)=1 pentru fiecare depozit și w(i, j)=1 pentru orice contract

fabrici.in	Iesire pe ecran (solutia nu este unica)
3 3 6 5 4 7 5 4 7 1 4 7 1 5 5 2 4 3 2 5 2 2 6 3 3 4 5 3 6 2	1 4 3 1 5 3 2 4 2 2 5 2 2 6 1 3 4 2 3 6 2