Subiectul 1

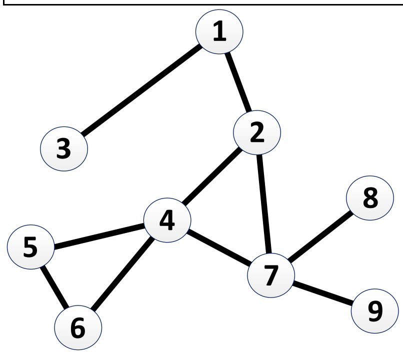
Se dă un graf neorientat conex cu n>3 vârfuri şi m>n muchii. Să se afişeze punctele critice în care sunt incidente muchii critice. Pentru fiecare astfel de punct se va afişa numărul de muchii critice care sunt incidente în el şi numărul de componente biconexe care îl conțin, fără a memora componentele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

Complexitate O(m)

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

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

graf.in	Iesire pe ecran (nu neaparat in aceasta ordine)
9 10 1 2 1 3 2 4 2 7 4 7 4 5 4 6 5 6 7 8 7 9	Puncte critice cerute: 1: incidente 2 muchii critice este in 2 componente biconexe 2: incidente 1 muchii critice este in 2 componente biconexe 7: incidente 2 muchii critice este in 3 componente biconexe



Subiectul 2

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

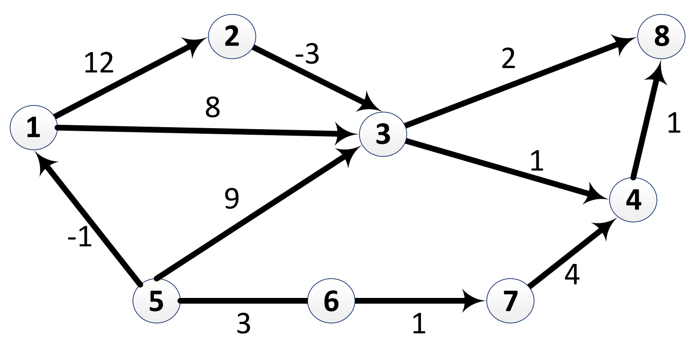
- Pe prima linie sunt două numere reprezentând numărul de vârfuri n (n>4) și numărul de arce m ale grafului
- Pe următoarele m linii sunt câte 3 numere întregi reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf (costul unui arc poate fi și **negativ**).
- Pe penultima linie este un nod sursa s
- Pe ultima linie sunt un număr natural k (0<k<n) reprezentând numărul de vârfuri destinație și k numere naturale t_1, t_2, ..., t_k reprezentând vârfuri destinație din G.

Spunem că un vârf y este accesibil din x în G dacă există un drum de la x la y. Presupunem că există cel puțin un vârf destinație care este accesibil din vârful sursă s.

a) Să se determine un vârf destinație care este cel mai depărtat de s, dar care este accesibil din s (un vârf destinație t pentru care $d(s,t) = \max\{d(s,t_i) | i=1,\dots, k, t_i$ accesibil din s$\}$). **Complexitate O(n+m)**

b) Pentru vârfurile s și t de la a) să se determine dacă există mai multe drumuri minime de la s la t. Dacă exista doar unul, se va afișa acest drum, dacă nu se vor afișa două dintre drumurile minime de la s la t . **Complexitate O(n+m)**

graf.in	Iesire pe ecran (nu este unică)
8 11 1 2 12 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 5 2 8 4	a) 8 b) 5 6 7 4 8 5 1 3 8



Explicații:

Sursa este 5, destinațiile sunt 8 și 4

d(5,8) = 9

d(5,4) = 8 => cea mai depărtată destinație de 5 este 8

Subiectul 3

Fişierul graf.in conţine următoarele informaţii despre un graf **bipartit** conex:

- pe prima linie sunt 2 numere naturale n şi m reprezentând numărul de vârfuri şi numărul de muchii
- pe următoarele m linii sunt perechi de numere x y (separate prin spaţiu) reprezentând extremităţile unei muchii

Se consideră graful G dat în fişierul graf.in. Notăm cu k numărul de vârfuri de grad impar din graf.

a) Folosind un algoritm de determinare a unui flux maxim într-o reţea de transport, determinaţi un cuplaj maxim în subgraful indus de mulţimea vârfurilor de grad impar din G.

b) Folosind punctul a) determinaţi dacă exista k/2 muchii care se pot elimina din G astfel încât să se obţină un graf cu următoarele proprietăţi:

- gradul fiecărui vârf din G' este egal cu cel din G sau cu unu mai mic.
- în G' în fiecare componentă conexă există câte un ciclu care conţine toate muchiile din componentă (o singura dată) **Complexitate $O(nm^2)$**

graf.in	Iesire pe ecran (solutia nu este unica)
8 9 1 5 1 6 1 7 2 5 3 5 3 7 3 4 8 7 8 4	1 6 2 5 3 7