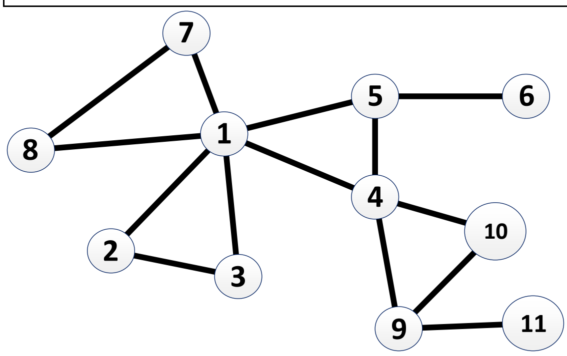
Subiectul 1

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 și 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:

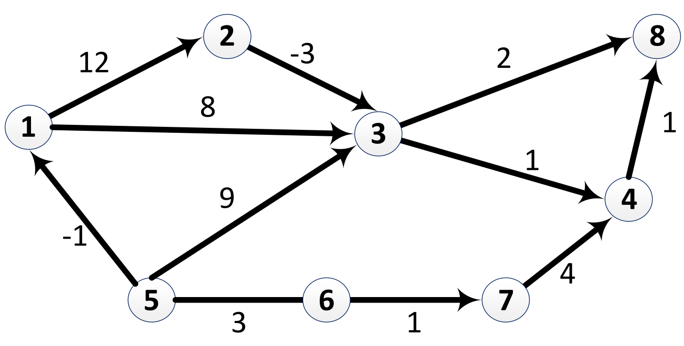
- Pe prima linie sunt două numere reprezentând numărul de vârfuri n (n>4) și numărul de arce m ale grafului
- Pe următoarele m linii sunt câte 3 numere întregi reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf (costul unui arc poate fi și **negativ**).
- Pe penultima 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 $d(s,t) = \max\{d(s,t_i) | i=1,\ldots, k,\ t_i$ 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

a) Se dau un număr natural n și două șiruri de n numere naturale s_in și s_out. Folosind algoritmul de determinare a unui flux maxim într-o rețea de transport, să se determine, dacă există, un graf orientat G cu secvența gradelor de intrare s_in și cu secvența gradelor de ieșire s_out. Se vor afișa arcele grafului dacă acesta există, și un mesaj corespunzător altfel.
b) În cazul în care graful cerut la G nu există, să determine dacă există doua numere i, j cuprinse între 1 și n (nu neparat distincte) astfel încât se poate construi un graf G’ cu secvența gradelor de intrare egală cu șirul obținut din s_in scăzând 1 din elementul i, și cu secvența gradelor de ieșire obținută din s_out scăzând 1 din elementul j. Se vor afișa arcele grafului G' dacă acesta există, și un mesaj corespunzător altfel.
c) În cazul în care graful cerut la G nu există, determinați dacă există un multigraf orientat G cu secvența gradelor de intrare s_in și cu secvența gradelor de ieșire s_out fără bucle (arce cu extremitățile egale).
Secvențele s_in și s_out se vor citi din fișierul secvente.in cu următoarea structură: pe prima linie este n, pe a doua linie elementele lui s_in separate prin spațiu, iar pe a treia linie elementele lui s_out separate prin spațiu.
Complexitate $O(mn^2)$, unde m este suma numerelor din s_in

secvente.in	Iesire pe ecran (solutia nu este unica)
3 1 0 3 2 2 0	a) nu exista b) 1 3 2 1 2 3 (i=3,j=1) c) 1 3 1 3 2 1 2 3