

Subiectul 1

Se dă un graf neorientat conex cu $n > 3$ vârfuri, m muchii, $m > n$ și un vârf s .

Să se afișeze muchiile a doi arbori parțiali ai grafului, T_1 și T_2 , dintre care unul, T_1 , este arbore de distanțe față de s ($d_{T_1}(s, u) = d_G(s, u)$ pentru orice vârf u din G), iar celălalt, T_2 , nu este arbore de distanțe față de s . Se va afișa în plus un vârf u pentru care $d_{T_2}(s, u) \neq d_G(s, u)$.

Complexitate $O(m)$

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

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

($d_G(x, y)$ = distanța de la x la y în G)

<i>graf.in</i>	<i>iesire pe ecran (soluția nu este unică)</i>
4 5	T1:
1 2	1 2
1 3	1 3
2 3	2 4
2 4	T2:
3 4	1 2
1	2 3
	2 4
	$u = 3$

Subiectul 2

Se citesc informații despre un graf **orientat** ponderat 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 **pozitive** reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf
- pe următoarea linie (a $(m+2)$ -a linie) din fișier este un număr natural k ($0 < k < n$) reprezentând numărul de vârfuri sursă; vârfurile sursă din G vor fi $1, 2, \dots, k$
- pe ultima linie a fișierului sunt două vârfuri t_1 și t_2 , reprezentând vârfurile destinație ale grafului.

Notăm cu $S = \{1, \dots, k\}$ mulțimea vârfurilor sursă din G și cu $T = \{t_1, t_2\}$ mulțimea vârfurilor 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 dintr-un vârf sursă.

Să se determine distanța între cele două mulțimi:

$$d(S, T) = \min \{d(x, y) \mid x \in S, y \in T\}$$

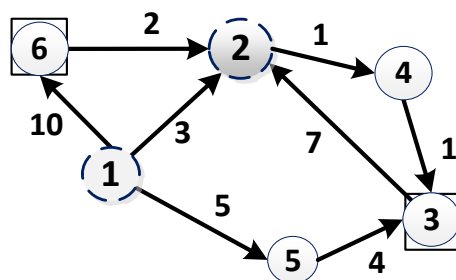
Să se determine în plus și o pereche de vârfuri (s, t) cu $s \in S$ și $t \in T$ cu

$$d(s, t) = d(S, T) = \min \{d(x, y) \mid x \in S, y \in T\}$$

și să se afișeze (pe ecran) un drum minim de la s la t . **Complexitate $O(m \log(n))$**

Exemplu

graf.in	Iesire pe ecran
<pre> 6 8 1 2 3 1 6 10 6 2 2 2 4 1 4 3 1 5 3 4 1 5 5 3 2 7 2 3 6 </pre>	<pre> distanța între multimi = 2 s=2 t=3 drum minim 2 4 3 </pre>



Explicații

$k=2 \Rightarrow S = \{1, 2\}$

$T = \{3, 6\}$

$d(1,3)=5, d(2,3)=2$

$d(1,6)=10, d(2,6)=\infty$

Cea mai mică este $d(2,3)$

Un drum minim de la 2 la 3 este 2 4 3

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 două numere i, j cuprinse între 1 și n (nu neapărat 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	a)
1 0 3	nu exista
2 2 0	b)
	1 3
	2 1
	2 3
	(i=3,j=1)
	c)
	1 3
	1 3
	2 1
	2 3