## 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, T1 și T2, dintre care unul, T1, este arbore de distante față de s ($d_{T1}(s, u) = d_G(s, u)$ pentru orice vârf u din G), iar celălalt, T2, nu este arbore de distanțe față de s. Se va afișa în plus un vârf u pentru care $d_{T2}(s, u) \neq d_G(s, u)$.
**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 vârful s

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

| ***graf.in*** | ***Iesire pe ecran (solutia nu este unica)*** |
|---|---|
| 4 5<br>1 2<br>1 3<br>2 3<br>2 4<br>3 4<br>1 | T1:<br>1 2<br>1 3<br>2 4<br>T2:<br>1 2<br>2 3<br>2 4<br>u = 3 |

## Subiectul 2

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

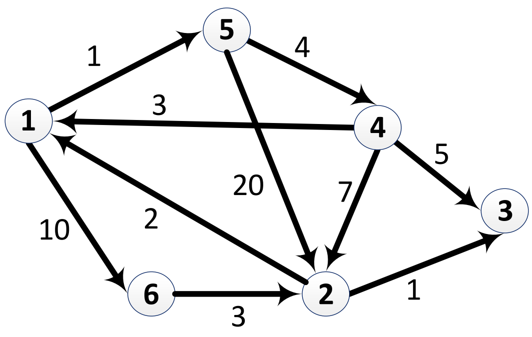
- Pe prima linie sunt două numere reprezentând numărul de vârfuri n (n>4) și numărul de arce m ale grafului, **m>n**
- Pe următoarele m linii sunt câte 3 numere întregi **pozitive** reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf
- Pe penultima linie este un număr natural b
- Pe ultima linie este un număr s reprezentând un nod sursă în graf.

În punctul s se află un călător care are bugetul b.

a) Să se determine un cel mai depărtat nod v din graf la care călătorul poate ajunge din s printr-un drum (elementar) de cost cel mult b, cât să se încadreze în buget (acel vârf pentru care se obține $\max\{d(s,u) | d(s,u) \leq b, \ u \text{ vârf în } V\}$ și să se afișeze un drum de cost minim de la s la v. Dacă sunt mai multe astfel de noduri se va alege cel cu indicele cel mai mic.

b) Observând că un circuit este format totuși dintr-un drum și un arc, călătorul va mai roagă să determinați în plus dacă poate face un traseu de cost cel mult b care pornește din s și se termina tot în s fără a trece de mai multe ori prin același vârf, altfel spus să determinați dacă există un circuit elementar în G de cost mai mic sau egal cu b care conține s și, în caz afirmativ, să afișați un astfel de circuit. **Complexitate O(mlog(n))**

| graf.in | Iesire pe ecran |
|---|---|
| 6 10<br>1 5 1<br>1 6 10<br>2 1 2<br>4 1 3<br>5 2 20<br>5 4 4<br>4 2 7<br>4 3 5<br>2 3 1<br>6 2 3<br>11<br>1 | a)<br>v=3<br>1 5 4 3<br>b)<br>1 5 4 1 |




d(1, 2) = 12
d(1, 3) = 10
d(1, 4) = 5
d(1, 5) = 5
d(1, 6) = 1
d(1, 7) = 10
b = 11 => cele mai mari distanțe mai mici sau egale cu 11 sunt d(1, 3) și d(1, 7)

## 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<br>1 0 3<br>2 2 0 | a)<br>nu exista<br>b)<br>1 3<br>2 1<br>2 3<br>(i=3,j=1)<br>c)<br>1 3<br>1 3<br>2 1<br>2 3 |