

Subiectul 1 (3 puncte)

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

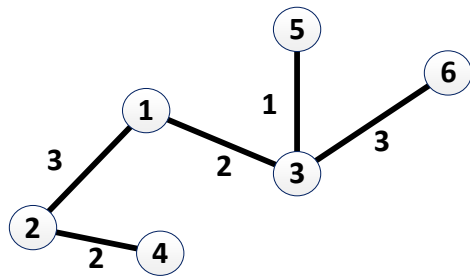
Informațiile despre graf se citesc din fișierul **graf.in** cu structura:

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

Pentru un lanț P în G definim ponderea lanțului P ca fiind produsul ponderilor muchiilor care îl compun.

Dacă G este arbore, să se afișeze pentru fiecare vârf v ponderea unicului lanț elementar de la s la v (sub forma v : pondere lanț de la s la v), altfel să se afișeze un arbore parțial al componente care conține s . **Complexitate $O(n+m)$**

| graf.in | iesire pe ecran |
|---|---|
| 6 5 1 2 3 1 3 2 2 4 2 3 5 1 3 6 3 1 | Este arbore 1: 0 2: 3 3: 2 4: 6 5: 2 6: 6 |



Subiectul 2 (3 puncte)

Se citesc informații despre un graf **neorientat** ponderat conex 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 muchii m ale grafului, $m > n$
- pe următoarele m linii sunt câte 3 numere pozitive reprezentând extremitatea inițială, extremitatea finală și costul unei muchii din graf

a) Să se afișeze costul unui arbore parțial de cost minim în G. **Complexitate $O(m \log(n))$.**

b) Se citesc de la tastatură două muchii **noi** date tot prin extremitatea inițială, extremitatea finală și cost. Știind că **doar una** dintre aceste muchii se va adăuga la graful G, decideți pe care o adăugați astfel încât noul graf să aibă un arbore parțial de cost minim cu cost cât mai mic și afișați muchiile unui arbore parțial de cost minim în acest graf. **Complexitate $O(n)$**

Exemplu

| <code>graf.in</code> | Iesire pe ecran (nu conteaza ordinea în care sunt afisate muchiile) |
|---|---|
| 5 5 1 2 1 1 4 2 2 3 4 3 4 8 4 5 6 Intrare de la tastatura 3 5 5 1 3 5 | a) 13 b) adaugam 3 5 1 2 1 4 2 3 3 5 |



Subiectul 3 (3 puncte)

Propuneți un **algoritm bazat pe algoritmul Ford-Fulkerson / Edmonds Karp** pentru rezolvarea următoarei probleme.

Pentru n proiecte, numerotate $1, \dots, n$ s-au înscris m studenți numerotați $1, \dots, m$, fiecare student depunând o listă de opțiuni cu proiectele la care vrea să participe.

- Dat un număr k de la tastatură, să se determine o listă de k asocieri proiect – student prin care k studenți diferiți sunt asociați la k proiecte diferite **Complexitate $O(km)$**
- Să se determine, dacă există, o modalitatea de a asocia toți studenții la proiecte astfel încât un student să fie asociat la exact 2 proiecte, iar la un proiect să fie asociați exact 2 studenți și să se afișeze o astfel de modalitate sub forma prezentată în exemplul de mai jos. Altfel se va afișa mesajul “nu este posibil”. **Complexitate $O(nm)$**

Datele despre proiecte și studenți se vor citi dintr-un fișier cu următoarea structură:

- pe prima linie sunt numerele naturale n și m
- pe următoarele linii sunt perechi de numere naturale i, j cu $i \in \{1, \dots, n\}$ și $j \in \{1, \dots, m\}$ cu semnificația: studentul j s-a înscris la proiectul i .

| graf.in | lesire pe ecran (solutia nu este unica) |
|--|---|
| 4 4 | a) |
| 1 1 | pentru $k=2$ |
| 1 2 | asocieri proiect - student |
| 1 3 | 1 1 |
| 2 1 | 2 2 |
| 2 2 | b) |
| 3 1 | asocieri proiect-student |
| 3 3 | 1 1 |
| 3 4 | 1 2 |
| 4 3 | 2 1 |
| 4 4 | 2 2 |
| | 3 3 |
| (primul este indicele proiectului, al doilea al studentului) | 3 4 |
| | 4 3 |
| | 4 4 |

