

Laborator 8

Exerciții

1. Grade¹

Se dă lista arcelor unui graf orientat. Să se determine nodurile care au gradul exterior egal cu gradul interior.

Date de intrare

Programul citește de la tastatură numărul N de noduri și numărul M de arce, iar apoi lista arcelor, formată din m perechi de forma $i \ j$, cu semnificația că există arc orientat de la i la j .

Date de ieșire

Programul va afișa pe ecran numărul C , reprezentând numărul de noduri care au gradul interior egal cu cel exterior, iar pe linie următoare afișează aceste noduri, ordonate crescător, separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 100.$$

2. Cunoscuți²

Într-un grup sunt N persoane, numerotate de la 1 la N și o persoană poate cunoaște alte persoane – relație care nu este reciprocă. Să se determine persoana cea mai cunoscută.

Date de intrare

Programul citește de la tastatură numărul N de noduri și numărul M de relații dintre persoane, iar apoi m perechi de forma $i \ j$, cu semnificația că persoana i cunoaște persoana j .

Date de ieșire

Programul va afișa pe ecran numărul de ordine al celei mai cunoscute persoane. Dacă răspunsul constă din mai multe persoane, se vor afișa toate, în ordine crescătoare a numerelor de ordine, separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 100.$$

¹ <https://www.pbinfo.ro/probleme/573/grade2>

² <https://www.pbinfo.ro/probleme/575/cunoscuti>

3. AfisCircuite³

Se dă lista arcelor unui graf orientat. Să se afișeze, în ordine lexicografică, toate circuitele de lungime trei.

Date de intrare

Programul citește de la tastatură numărul N de noduri și numărul M de arce, iar apoi lista arcelor, formată din m perechi de forma $i \ j$, cu semnificația că există arc orientat de la nodul i la nodul j .

Date de ieșire

Programul va afișa pe ecran circuitele cerute, câte un circuit pe o linie a ecranului, nodurile unui circuit fiind separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 100.$$

4. Turneu⁴

Un graf orientat se numește graf turneu dacă oricare ar fi două noduri diferite i, j , între ele există un singur arc: arcul $(i \ j)$ sau arcul $(j \ i)$. În orice graf turneu există un drum elementar care trece prin toate nodurile grafului.

Se dă un graf turneu cu N noduri. Să se determine un drum elementar care să conțină toate nodurile grafului.

Date de intrare

Programul citește de la tastatură numărul de noduri N , iar apoi $\frac{n(n-1)}{2}$ perechi $i \ j$, cu semnificația că există arcul $(i \ j)$.

Date de ieșire

Programul va afișa pe ecran cele N noduri ale drumului determinat, separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 100.$$

Orice drum corect determinat este acceptat.

5. Drumuri⁵

Se dă un graf orientat cu N noduri și un nod p . Să se afișeze toate nodurile q ale grafului, diferite de p , cu proprietatea că există cel puțin un drum de la p la q și lungimea drumului minim este pară.

³ <https://www.pbinfo.ro/probleme/577/afiscircuite>

⁴ <https://www.pbinfo.ro/probleme/582/turneu>

⁵ <https://www.pbinfo.ro/probleme/581/drumuri>

Date de intrare

Programul citește de la tastatură numărul N de noduri, nodul p și numărul M de arce, iar apoi lista arcelor, formată din M perechi de forma $i \ j$, cu semnificația că există arc orientat de la nodul i la nodul j .

Date de ieșire

Programul va afișa pe ecran numărul C de noduri care respectă cerința, iar pe rândul următor cele C noduri, în ordine crescătoare, separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 100.$$

Prin drum minim se înțelege un drum de lungime minimă.

6. Mall⁶

Într-un mall sunt N centre comerciale, numerotate de la 1 la n , unite între ele prin coridoare unidirecționale. Să se determine, dacă există, un centru comercial în care se poate ajunge din oricare altul.

Date de intrare

Programul citește de la tastatură numărul N de centre comerciale și numărul M de perechi de centre unite prin coridoare, iar apoi lista acestora, formată din M perechi de forma $i \ j$, cu semnificația că există un coridor unidirecțional de la centrul i la centrul j .

Date de ieșire

Programul va afișa pe ecran centrul comercial determinat. Dacă există mai multe centre comerciale în care se poate ajunge din toate celelalte, se va afișa acela cu numărul de ordine mai mic. Dacă nu poate fi determinat nici un centru comercial în care se poate ajunge din toate celelalte, se va afișa mesajul NU EXISTA.

Restricții

$$1 \leq N \leq 100.$$

7. TopSort⁷

Se dă un graf orientat aciclic cu N noduri numerotate de la 1 la N . Să se realizeze o sortare topologică a nodurilor.

Date de intrare

Fișierul de intrare `topsort.in` conține pe prima linie numerele $N \ M$, reprezentând numărul de noduri și numărul de arce din graf, iar pe următoarele M linii câte o pereche de noduri $i \ j$, cu semnificația că în graf există arcul $(i \ j)$.

⁶ <https://www.pbinfo.ro/probleme/587/mall>

⁷ <https://www.pbinfo.ro/probleme/1861/topsort>

Date de ieșire

Fișierul de ieșire `topsort.out` va conține pe prima linie cele N noduri ale grafului, separate prin exact un spațiu, în ordinea dată de sortarea topologică.

Restricții

$$1 \leq N \leq 100.000, 1 \leq M \leq 400.000.$$

8. Coborare⁸

Un turist a făcut o excursie până în vârful unui munte. Acum, el dorește să coboare înapoi la cabană. Muntele este presărat cu poieni între care se află cărări. Fiecare cărare leagă două poieni diferite, iar într-o poiană pot ajunge mai multe cărări. Turistul se întreabă: câte trasee diferite există din vârful muntelui la cabană, care să meargă numai pe cărări și numai la vale?

Date de intrare

Fișierul de intrare `coborare.in` conține pe prima linie patru numere $N \ M \ V \ C$, unde

- N este numărul de poieni;
- M este numărul de cărări;
- V este poiana din vârful muntelui (de unde pornește turistul);
- C este poiana în care se află cabana.

Pe următoarele M linii se află câte o pereche de numere $x \ y$, cu semnificația că există o cărare care coboară din poiana x în poiana y .

Date de ieșire

În fișierul de ieșire `coborare.out` se va scrie un singur număr, respectiv numărul de trasee distincte de la V la C , modulo 100003 .

Restricții

$$1 \leq N \leq 100.000, 1 \leq M \leq 300.000, 1 \leq V, C \leq N.$$

Poienile au numere între 1 și N .

9. Orașe⁹

Într-o țară oarecare sunt N orașe, numerotate de la 1 la n , între care există șosele unidirecționale. Președintele țării vrea să facă o vizită electorală prin mai multe orașe în felul următor: aterizează cu elicopterul în orașul p , merge până în orașul q folosind rețeaua de șosele existentă, de unde pleacă cu elicopterul. Din punct de vedere politic unele orașe sunt sigure (conduse de aliați politici ai președintelui), altele sunt nesigure (conduse de adversari), iar președintele dorește să parcurgă numai orașe sigure.

Consilierii președintelui i-au propus mai multe perechi de orașe $p \ q$ care pot fi alese ca punct de plecare și de sosire pentru vizită. Președintele vă cere să verificați pentru fiecare pereche dată dacă există un drum care să treacă numai prin orașe sigure.

⁸ <http://varena.ro/problema/coborare>

⁹ <https://www.pbinfo.ro/probleme/585/orase>

Date de intrare

Programul citește de la tastatură numerele N M k v , reprezentând:

N – numărul total de orașe,

M – numărul de perechi de orașe între care există șosea unidirecțională,

k – numărul de orașe sigure,

v – numărul de perechi de orașe propuse pentru vizită.

Programul va citi apoi M perechi de orașe i j , cu semnificația că există șosea unidirecțională de la orașul i la orașul j . Apoi se vor citi cele k orașe sigure, apoi cele v perechi de orașe candidat p q .

Date de ieșire

Programul va afișa pe ecran pe v linii mesajul DA sau NU, după cum perechea corespunzătoare de orașe candidat poate fi aleasă pentru vizită, sau nu.

Restricții

$$1 \leq k \leq N \leq 100$$

O pereche p q poate fi aleasă chiar dacă orașul p sau orașul q sunt nesigure. Este important să fie sigure orașele intermediare.

O pereche p q poate fi aleasă dacă există șosea de la p la q .

10. Project management¹⁰

La o firmă de software se lucrează la un mare proiect. Proiectul constă în executarea a N (număr natural) faze de dezvoltare, numerotate cu numerele $1, 2, \dots, n$. Unele faze pot fi executate în paralel (în același timp), însă executarea altor faze nu poate fi începută până când nu se finalizează executarea anumitor faze.

Să se scrie un program care să determine:

a) timpul minim T în care se poate finaliza executarea proiectului.

b) pentru fiecare fază k (k din $\{1, 2, \dots, n\}$), momentul de timp c_k la care poate începe faza k cel mai devreme, respectiv momentul de timp d_k la care poate începe faza k cel mai târziu, fără a influența durata totală de executare a proiectului.

Date de intrare

Fișierul de intrare `pm.in` conține:

- pe prima linie, un număr natural N , reprezentând numărul fazelor proiectului;
- pe a doua linie, N numere naturale, separate prin câte un spațiu, reprezentând timpul necesar finalizării fiecărei faze;
- pe fiecare linie k dintre următoarele n linii, un număr natural m_k și un șir a format din m_k numere naturale: $a_1, a_2, \dots, a_{m_k}$, cele $m+1$ numere din linie fiind separate prin câte un

¹⁰ <https://www.pbinfo.ro/probleme/1101/project-management>

Algoritmica grafurilor

spațiu, m_k reprezentând numărul de faze ce trebuie finalizate înaintea începerii fazei k , iar numerele din șirul a reprezentând numerele de ordine ale fazelor ce trebuie finalizate înaintea începerii fazei k .

Date de ieșire

Fișierul de ieșire **pm.out** va conține $N+1$ linii. Pe prima linie se va scrie numărul natural T , iar pe fiecare linie k dintre următoarele N linii, se vor scrie cele două numere naturale c_k și d_k , separate prin câte un spațiu.

Restricții

$$1 \leq N \leq 100$$

Timpul necesar finalizării executării oricărei faze nu va depăși $1.000.000$

Se consideră că executarea proiectului începe la momentul de timp 0 .

Nu vor exista dependențe circulare (proiectul întotdeauna se poate finaliza).