

Laborator 3

Exerciții

1. DFS¹

Se consideră un graf neorientat cu N vârfuri și M muchii și un vârf X .

Să se afișeze vârfurile vizitate în urma parcurgerii în adâncime (*Depth First Search*) a grafului, pornind din vârfurile X .

Date de intrare

Fișierul de intrare `dfs.in` conține pe prima linie trei numere naturale N M X , având următoarea semnificație: N este numărul de noduri, M este numărul de muchii din graf, iar X este vârful din care se începe parcurgerea. Următoarele M linii conțin câte două numere i j cu semnificația că există muchie între i și j .

Date de ieșire

Fișierul de ieșire `dfs.out` va conține pe prima linie vârfurile vizitate în urma parcurgerii în adâncime a grafului, aceasta începând din vârful X .

Restricții

$$2 \leq N \leq 100, 1 \leq M \leq \frac{N(N-1)}{2}.$$

Dacă în timpul parcurgerii, la un moment dat vârful curent al grafului este Y , vecinii nevizitați ai lui Y se vor analiza în ordine crescătoare.

2. Conex²

Se dă lista muchiilor unui graf neorientat. Să se verifice dacă graful este sau nu conex.

Date de intrare

Fișierul de intrare `conex.in` conține pe prima linie numărul N , reprezentând numărul de vârfuri ale grafului. Fiecare dintre următoarele linii conține câte o pereche de numere i j , cu semnificația că există muchie între i și j .

¹ <https://www.pbinfo.ro/probleme/539/dfs>

² <https://www.pbinfo.ro/probleme/437/conex>

Date de ieșire

Fișierul de ieșire **conex.out** va conține mesajul DA, dacă graful dat este conex, respectiv NU, în caz contrar.

Restricții

$$1 \leq N \leq 100, 1 \leq i, j \leq N.$$

În fișierul de intrare muchiile se pot repeta.

3. Componente conex³

Se dă lista muchiilor unui graf neorientat. Să se afișeze componentele conex³ ale acestui graf.

Date de intrare

Fișierul de intrare **componenteconexe.in** conține pe prima linie numărul N , reprezentând numărul de vârfuri ale grafului. Fiecare dintre următoarele linii conține câte o pereche de numere i și j , cu semnificația că există muchie între i și j .

Date de ieșire

Fișierul de ieșire **componenteconexe.out** va conține pe prima linie numărul de componente conex³ nrc . Fiecare dintre următoarele nrc linii va conține în ordine crescătoare, separate printr-un spațiu, vârfurile din componenta conexă curentă.

Ordinea de afișare a componentelor conex³ va fi cea crescătoare a vârfului cu eticheta minimă din fiecare componentă.

Restricții

$$1 \leq N \leq 100, 1 \leq i, j \leq N.$$

În fișierul de intrare muchiile se pot repeta.

4. ComponenteConex⁴

Se dă lista muchiilor unui graf neorientat. Să se determine numărul de muchii care pot fi eliminate din graf astfel încât numărul de componente conex³ ale grafului să nu se modifice.

Date de intrare

Fișierul de intrare **componenteconexe2.in** conține pe prima linie numărul N , reprezentând numărul de vârfuri ale grafului. Fiecare dintre următoarele linii conține câte o pereche de numere i și j , cu semnificația că există muchie între i și j .

Date de ieșire

Fișierul de ieșire **componenteconexe2.out** va conține pe prima linie numărul NR de muchii care pot fi eliminate.

³ <https://www.pbinfo.ro/probleme/438/componenteconexe>

⁴ <https://www.pbinfo.ro/probleme/537/componenteconexe2>

Restricții

$$1 \leq N \leq 100, 1 \leq i, j \leq N.$$

În fișierul de intrare muchiile se pot repeta.

5. Retea⁵

Se consideră o rețea formată din N servere, numerotate de la 1 la N . În rețea există M perechi de servere $x \ y$ cunoscute între care există legături de comunicație directe. Între oricare două servere din rețea există legături, fie directe, fie prin intermediul altor servere.

Stabiliți pentru fiecare dintre cele N servere dacă eliminarea sa din rețea conduce la pierderea legăturii dintre cel puțin două servere rămase.

Date de intrare

Fișierul de intrare `retea.in` conține pe prima linie numerele N și M . Pe următoarele M linii se vor afla câte două numere naturale $x \ y$, reprezentând perechi de servere între care există legături directe.

Date de ieșire

Fișierul de ieșire `retea.out` va conține pe prima linie N numere naturale 0 sau 1:

- al k -lea număr va fi 0 dacă prin eliminarea serverului cu numărul k rămân legături între oricare două servere rămase
- al k -lea număr va fi 1 dacă prin eliminarea sa se pierde legătura între cel puțin două servere rămase.

Restricții

$$1 \leq N \leq 100, 1 \leq x, y \leq N.$$

6. Arrows⁶

"Arrows" este un joc care se joacă pe o tablă dreptunghiulară a cărei suprafață este împărțită în $N \times M$ celule, aranjate pe N linii și M coloane. În fiecare celulă se află o săgeată (*sus*, *jos*, *stânga* sau *dreapta*), ca în figura de mai jos.

←	→	→	↓	↑
→	↑	↓	←	→
↓	↑	→	→	↓
→	↑	←	←	←
←	→	↓	↓	↓
↑	↑	←	↓	↑

⁵ <https://www.pbinfo.ro/probleme/1707/retea>

⁶ <https://www.pbinfo.ro/probleme/1049/arrows>

Când este la mutare, un jucător poate alege o *poziție de start* pe care plasează un jeton, apoi deplasează jetonul la celula învecinată în sensul indicat de săgeată. Deplasarea continuă până când jetonul părăsește tabla de joc, caz în care jucătorul obține un punctaj egal cu numărul de celule parcurse de jetonul său.

Există însă poziții de start denumite *favorabile*, pentru care jetonul nu va părăsi niciodată tabla de joc. De exemplu, toate pozițiile din figură cu fundal gri sunt *favorabile*. Jucătorul care alege o poziție de start *favorabilă* obține un punctaj egal cu numărul de celule distincte vizitate înmulțit cu 1000.

Scrieți un program care, cunoscând configurația tablei de joc, rezolvă una dintre următoarele cerințe:

- 1) determină punctajul pe care îl obține un jucător care plasează jetonul său pe o poziție de start specificată;
- 2) determină numărul de celule *favorabile* de pe tabla de joc;
- 3) determină punctajul maxim pe care jucătorul îl poate obține la o mutare, alegând convenabil poziția de start.

Date de intrare

Fișierul de intrare `arrows.in` conține pe prima linie cerința care trebuie să fie rezolvată (1, 2 sau 3). Pe a doua linie se află numerele naturale N și M , care reprezintă numărul de linii și respectiv de coloane de pe tabla de joc. Pe următoarele N linii se află câte M numere din mulțimea $\{1, 2, 3, 4\}$ reprezentând săgețile aflate în celulele de pe tabla de joc (1 semnificând *săgeata la dreapta*, 2 *săgeata în sus*, 3 *săgeata la stânga* și 4 *săgeata în jos*). Pe ultima linie sunt scrise numerele naturale `lin col`, reprezentând linia și coloana pe care se află poziția de start specificată. Valorile scrise pe aceeași linie în fișierul de intrare sunt separate prin spații.

Date de ieșire

Fișierul de ieșire `arrows.out` va conține o singură linie pe care va fi scris un număr natural reprezentând răspunsul pentru cerința specificată pe prima linie a fișierului de intrare.

Restricții

$$1 \leq N, M \leq 500$$

Liniile sunt numerotate de la 1 la N , iar coloanele de la 1 la M .

7. Dirty⁷

NSA a înființat *Departamentul pentru Interceptarea Rețelelor Teroriste* pe *Yahoo*, *Facebook*, *Apple*, *Google* și *Skype* (NSA PRISM). Departamentul veghează la liniștea noastră, a muritorilor simpli fără nimic de ascuns, și analizează datele obținute folosind o rețea de calcul formată din calculatoare conectate prin cabluri bidirecționale. Fiecare cablu leagă exact două calculatoare, dar un calculator poate avea mai multe cabluri. Oricare două calculatoare sunt legate

⁷ <http://varena.ro/problema/dirty>

Algoritmica grafurilor

direct prin cel mult un cablu. În prezent, rețeaua este conexă, adică oricare două calculatoare pot comunica, direct sau indirect.

Anarhista Julianna Sage și-a folosit puterea de seducție pentru a îl convinge pe tehnicianul Steward Noden să-i ofere un tur al laboratorului. În realitate, ea ține pe stick un virus cu care poate distruge orice calculator din rețea. Pentru a sabota cât mai mult rețeaua, Julianna dorește să viruseze acel calculator prin dispariția căruia rețeaua se fragmentează în cât mai multe sub-rețele deconectate una de alta.

Cunoscând structura rețelei, ajutați-o pe Julianna să afle toate variantele de atac pe care le are.

Date de intrare

Fișierul de intrare `dirty.in` conține, pe prima linie, numărul N de calculatoare și numărul M de cabluri. Pe următoarele M linii sunt indicate conexiunile, sub forma câte unei perechi $X \ Y$ cu semnificația că există un cablu între calculatoarele numerotate X și Y .

Date de ieșire

În fișierul de ieșire `dirty.out` se vor scrie, pe prima linie, R = numărul de sub-rețele în care poate fragmenta Julianna rețeaua și V = numărul de variante de a sabota un calculator pe care le are ea. Pe a doua linie se vor scrie cele V numere de ordine ale acestor calculatoare, în ordine crescătoare și despărțite prin spații.

Restricții

$$1 \leq N \leq 100.000, 1 \leq M \leq 300.000, 1 \leq X, Y \leq N$$

Julianna garantează că există un calculator prin dispariția căruia rețeaua să se fragmenteze.

8. Parțial⁸

Se dă un graf neorientat conex cu N vârfuri și număr par de muchii. Să se determine un graf parțial al celui dat care să fie conex și să fie obținut prin eliminarea a jumătate din numărul de muchii.

Date de intrare

Fișierul de intrare `partial.in` conține pe prima linie numărul N , iar pe următoarele linii perechi de numere $i \ j$, cu semnificația că există muchie de la i la j .

Date de ieșire

Fișierul de ieșire `partial.out` va conține matricea de adiacență a grafului parțial obținut, câte o linie a matricei pe o linie a fișierului, elementele fiecărei linii fiind separate prin exact un spațiu.

Restricții

$$1 \leq N \leq 200, 1 \leq i, j \leq N.$$

Se garantează existența unui graf parțial cu proprietatea cerută.

⁸ <https://www.pbinfo.ro/probleme/544/partial>

9. Bipartit⁹

Se dă lista muchiilor unui graf neorientat cu N vârfuri, etichetate de la 1 la N . Să se verifice dacă graful este bipartit.

Date de intrare

Fișierul de intrare `bipartit1.in` conține pe prima linie numerele N și M , reprezentând numărul de vârfuri ale grafului și numărul de muchii. Fiecare dintre următoarele M linii conține câte o pereche de numere $i \ j$, cu semnificația că există muchie între i și j .

Date de ieșire

Fișierul de ieșire `bipartit1.out` va conține pe prima linie mesajul DA, dacă graful este bipartit, respectiv NU în caz contrar.

Dacă mesajul este DA, următoarele două linii vor conține două mulțimi care formează partiția vârfurilor. Elementele fiecărei mulțimi vor fi afișate în ordine crescătoare, separate prin exact un spațiu. Prima mulțime va fi cea care conține vârful 1.

Restricții

$$1 \leq N \leq 15, 1 \leq i, j \leq N.$$

10. Componente biconexe¹⁰

Un graf neorientat se numește *biconex* dacă este conex și nu conține puncte de articulație.

Un *punct de articulație* este un nod care prin ștergerea sa și a muchiilor incidente cu el implică pierderea conexității.

O *componentă biconexă* a unui graf neorientat conex reprezintă un subgraf maximal biconex.

Se numește *muchie critică* într-un graf neorientat conex o muchie care prin eliminarea ei duce la pierderea conexității grafului.

Dându-se un graf neorientat conex, sa se găsească¹¹:

1. Componentele biconexe;
2. Punctele de articulație;
3. Muchiile critice.

Date de intrare

Fișierul de intrare `componentebiconexe.in` va conține pe prima linie un număr p , care poate lua valorile 1, 2 sau 3. Pe a doua linie se află N și M (numărul de noduri, respectiv numărul de muchii ale grafului dat). Pe următoarele M linii se află câte o pereche de numere $a \ b$ cu semnificația ca există muchie între nodurile a și b .

⁹ <https://www.pbinfo.ro/probleme/472/bipartit1>

¹⁰ <https://www.pbinfo.ro/probleme/1572/componentebiconexe>

¹¹ <https://www.pbinfo.ro/articole/5983/biconexitate>

Date de ieşire

Fişierul de ieşire `componentebiconexe.out` va conţine răspunsul la una dintre cerinţe astfel:

- Dacă $p = 1$: pe prima linie se va afişa numărul de componente biconexe (nr). Pe următoarele nr linii se va afişa câte o componentă biconexă astfel: numărul de noduri din componenta biconexă urmat de nodurile din această componentă, toate separate prin spaţiu.
- Dacă $p = 2$: pe prima linie se va afişa numărul de puncte de articulaţie. Pe a doua linie se vor afişa separate prin spaţiu toate punctele de articulaţie.
- Dacă $p = 3$: pe prima linie se va afişa numărul de muchii critice, apoi pe câte o linie vor fi afişate muchiile critice.

Restricţii

$$3 \leq N \leq 100.000, 3 \leq M \leq 200.000.$$

Componentele biconexe, nodurile dintr-o componentă biconexă, punctele de articulaţie şi muchiile critice se pot afişa în orice ordine.

În fişierul de intrare nu va exista aceeaşi muchie de mai multe ori sau muchii de la un nod la el însuşi.