

Laborator 7

Exerciții

1. Prim¹

Se dă un graf neorientat ponderat conex cu N vârfuri și M muchii – în care fiecare muchie are asociat un cost, număr natural strict pozitiv. Folosind algoritmul lui Prim, determinați un arbore parțial de cost minim, cu rădăcina în vârful 1.

Date de intrare

Fișierul de intrare `prim.in` conține pe prima linie numerele n m , iar următoarele linii câte un triplet i j c , cu semnificația: există muchia (i, j) și are costul c .

Date de ieșire

Fișierul de ieșire `prim.out` va conține pe prima linie costul arborelui de cost minim determinat, iar pe a doua linie vectorul de tați al arborelui, cu elementele separate prin exact un spațiu.

Restricții

$1 \leq n \leq 100$.

Costul unei muchii va fi mai mic decât 1.000.

2. Kruskal²

Se dă un graf neorientat ponderat conex cu N vârfuri și M muchii – în care fiecare muchie are asociat un cost, număr natural strict pozitiv. Folosind algoritmul lui Kruskal, determinați un arbore parțial de cost minim.

Date de intrare

Fișierul de intrare `kruskal.in` conține pe prima linie numerele n m , iar următoarele linii câte un triplet i j c , cu semnificația: există muchia (i, j) și are costul c .

Date de ieșire

Fișierul de ieșire `kruskal.out` va conține pe prima linie costul arborelui de cost minim determinat, iar pe următoarele $n-1$ linii câte o pereche de numere i j , cu semnificația că muchia (i, j) aparține arborelui parțial de cost minim determinat.

¹ <https://www.pbinfo.ro/probleme/590/prim>

² <https://www.pbinfo.ro/probleme/592/kruskal>

Restricții

$$1 \leq n \leq 100.$$

Costul unei muchii va fi mai mic decât 1.000.

3. Zăpada³

Orașul Kruskal are N intersecții unite prin M străzi bidirecționale. Datorită ninsorii de peste noapte, străzile sunt acoperite cu zăpadă. Administratorul orașului, Gigel, a determinat cu mari eforturi pentru fiecare stradă costul deszăpezirii ei și acum dorește deszăpezirea unor străzi astfel încât costul total al deszăpezirii lor să fie minim, și să se poată circula între oricare două intersecții pe străzi deszăpezite.

Maleficul Costel îl forțează pe Gigel să deszăpezească anumite străzi, din motive doar de el știute. Ajutați-l pe Gigel să determine costul minim pentru deszăpezirea orașului, astfel încât să fie deszăpezite străzile dorite de Costel și să se poată circula între oricare două intersecții pe străzi deszăpezite.

Date de intrare

Fișierul de intrare **zapada.in** conține pe prima linie numerele N M , iar pe următoarele M linii câte un triplet i j c , cu semnificația: există stradă între intersecțiile i j și are costul deszăpezirii egal cu c . Urmează o linie care conține numărul k de străzi pe care Costel le dorește deszăpezite, iar următoarea linie conține k numere distincte între 1 și M , reprezentând numărul de ordine al străzilor care vor fi obligatoriu deszăpezite.

Date de ieșire

Fișierul de ieșire **zapada.out** va conține pe prima linie costul total minim al deszăpezirii orașului, în condițiile precizate.

Restricții

$$1 \leq n \leq 100, 1 \leq k < m \leq \frac{n(n-1)}{2}$$

Costurile deszăpezirii străzilor sunt numere naturale nenule cel mult egale cu 1.000.

Numerele celor k străzi care trebuie deszăpezite sunt dintre numerele de ordine ale celor M străzi date.

4. ComponenteConexe⁴

Se consideră un graf neorientat cu N vârfuri și M muchii. Cele M muchii se elimină pe rând din graf.

Pentru fiecare muchie eliminată trebuie să spuneți câte componente conexe are graful.

³ <https://www.pbinfo.ro/probleme/616/zapada>

⁴ <https://www.pbinfo.ro/probleme/2282/componenteconexe4>

Date de intrare

Fișierul de intrare `componenteconexe4.in` conține pe prima linie o pereche de numere N și M , iar pe următoarele M linii se află câte două valori x și y , separate prin spațiu, reprezentând o muchie din graf. Muchiile se dau în ordinea în care ele se elimină din graf.

Date de ieșire

Fișierul de ieșire `componenteconexe4.out` va conține exact M linii, pe fiecare linie k aflându-se un singur număr natural reprezentând numărul componentelor conexe din graf după eliminarea celei de-a k -a muchii.

Restricții

$$1 \leq N \leq 100.000, 1 \leq M \leq 500.000.$$

Între două vârfuri va exista cel mult o muchie.

5. Cotoroanta⁵

Castelul Cotoroanței are N camere legate prin M coridoare. Un coridor leagă exact două camere și oricare două camere sunt legate prin cel mult un coridor. Se cunoaște lungimea fiecărui coridor (camerele au dimensiuni neglijabile). În fiecare din următorii K ani, Cotoroanța își extinde castelul cu câte o cameră. Noua cameră este legată prin coridoare de toate camerele existente.

De fiecare *Halloween*, Cotoroanța iluminează toate camerele castelului cu instalația electrică. Ea poate construi o instalație ramificată, dar are o singură baterie și dorește să cumpere cât mai puțin cablu electric. Să se calculeze, pentru castelul inițial și pentru următorii K ani, lungimea minimă de cablu necesară.

Date de intrare

Fișierul de intrare `cotoroanta.in` conține, pe prima linie, valorile N , M și K . Pe următoarele M linii apar triplete de numere x y z cu semnificația că există un coridor între camerele x și y de lungime z . Pe următoarele K linii sunt indicate lungimile noilor coridoare. Linia corespunzătoare anului i va conține $N + i - 1$ numere.

Date de ieșire

În fișierul de ieșire `cotoroanta.out` se vor scrie, în ordine cronologică, lungimile de cablu necesare pentru castelul inițial și pentru următorii K ani.

Restricții

$$1 \leq N \leq 5.000, 1 \leq M \leq 100.000, 1 \leq K \leq 100$$

Lungimile coridoarelor sunt numere întregi între 1 și 1.000.000.

6. Channels⁶

⁵ <http://varena.ro/problema/cotoroanta>

⁶ <http://varena.ro/problema/channels>

În Waterland, există n lacuri (numerotate de la 1 la n) și m canale care fac legătura între ele. Lățimea (în metri) a fiecărui canal este cunoscută. Pe canale se poate naviga în ambele direcții. Se știe că o barcă cu lățime de un metru poate naviga pe oricare canal.

Scrieți un program, ce calculează numărul minim de canale care ar trebui să fie lărgite, astfel încât o barcă cu lățime de k metri să poată face o excursie între fiecare două lacuri (barca poate trece de la un lac la altul dacă lățimea sa este mai mică sau egală cu lățimea canalului care leagă lacurile).

Date de intrare

Fișierul de intrare `channels.in` conține pe prima linie numerele întregi n și m .

Pe fiecare din următoarele m linii sunt date trei numere întregi i , j și w , care arată că există un canal de lățime w între lacurile i și j ($1 \leq i, j \leq n$).

Pe ultima linie este dat k , întreg.

Date de ieșire

Fișierul de ieșire `channels.out` va conține numărul minim de canale care ar trebui să fie lărgite.

Restricții

$$1 < n \leq 1.000, 1 < m \leq 100.000, 1 \leq w \leq 200, 1 \leq k \leq 200.$$

7. Flota⁷

Mândria complexului de agrement de la Șmenu sunt plimbările cu vaporul pe salba de N lacuri legate prin M canale. Un vapor de lățime L poate naviga între două lacuri dacă există o cale între cele două lacuri formată numai din canale de lățime cel puțin egală cu L . Proprietarul complexului, Trăiam Binescu, dorește să cumpere o flotă de vapoare, toate identice, care să deservească toate lacurile. Deoarece prețurile variază pentru vapoare de diverse lățimi, Binescu își notează K lățimi și dorește să afle, în fiecare caz, numărul minim de vapoare necesare.

Date de intrare

Fișierul de intrare `flota.in` conține, pe prima linie, numerele N , M și K .

Următoarele M linii conțin triplete de numere x y w , semnificând că între lacurile x și y există un canal de lățime w .

Ultimele K linii conțin câte un număr L_i , semnificând o posibilă lățime a vapoarelor de cumpărat.

Date de ieșire

În fișierul de ieșire `flota.out` se vor scrie K linii. Linia i va conține un singur număr, respectiv numărul de vapoare care sunt necesare pentru lățimea L_i . Răspunsurile trebuie afișate în aceeași ordine cu întrebările.

⁷ <http://varena.ro/problema/flota>

Restricții

$$1 \leq N \leq 50.000, 1 \leq M \leq 1.000.000, 1 \leq K \leq 100.000$$

Între oricare două lacuri există cel mult un canal. Toate canalele au dublu sens.

Lățimile canalelor și ale vapoarelor sunt întregi, pozitive, cel mult egale cu 50.000.

8. Meta⁸

Elevii de la cercul de informatică înțeleg atât de bine algoritmii pentru găsirea arborelui parțial de cost minim, încât îi pot implementa și cu ochii închiși. De aceea, profesorul le-a dat o problemă ceva mai grea: găsirea celui de-al doilea arbore parțial în ordinea costului.

Se dă un graf neorientat conex cu N noduri și M muchii. Fiecare muchie are un cost pozitiv asociat și toate muchiile au costuri diferite. Ajutați-i pe elevii de la cerc să calculeze costul celui de-al doilea arbore parțial minim. Mai exact, se cere cel mai mic număr natural K cu proprietățile:

- Există un arbore parțial de cost total K .
- K este strict mai mare decât costul arborelui parțial de cost minim pentru graful dat.

Date de intrare

Fișierul de intrare `meta.in` conține pe prima linie numerele N și M , despărțite printr-un spațiu. Pe următoarele M linii se găsește câte o pereche de numere x y c cu semnificația că între nodurile x și y există o muchie de cost c .

Date de ieșire

În fișierul de ieșire `meta.out` se va tipări doar numărul K .

Restricții

$$1 \leq N \leq 1.000, 1 \leq M \leq 300.000, M \geq N.$$

Nodurile sunt numerotate de la 1 la N .

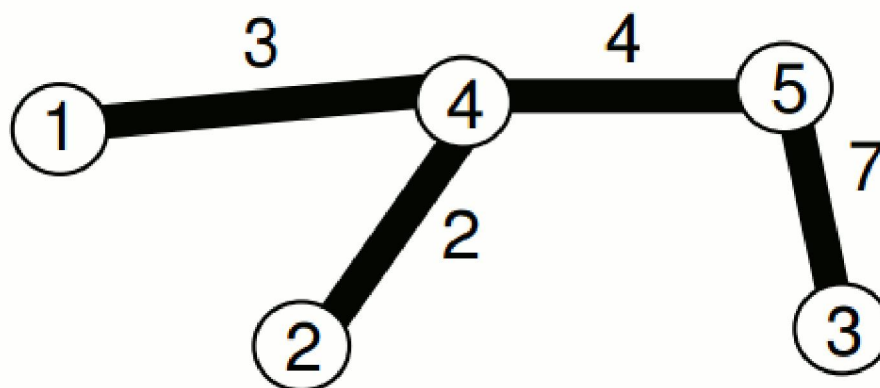
Costurile muchiilor sunt numere naturale distincte cuprinse între 1 și 1.000.000.

9. Recon⁹

Tinerii programatori Peter și Stancho au fost angajați de două agenții spațiale. Agenția lui Peter a proiectat o nouă stație spațială compusă din N module, numerotate de la 1 la N . Unele module sînt legate prin coridoare în așa fel încît se poate ajunge de la de la oricare modul la oricare alt modul printr-o cale unică de coridoare (vezi figura). Lungimea fiecărui coridor este un întreg pozitiv. Există cel mult un coridor între oricare două module.

⁸ <http://varena.ro/problema/meta>

⁹ <http://varena.ro/problema/recon>



Șefii lui Peter ar dori să țină secret proiectul. De aceea Peter a codat topologia stației dând pentru fiecare două module distanța dintre ele (adică suma lungimilor coridoarelor pe calea unică dintre module).

Dar Stancho are o problemă grea - el a promis șefilor săi că va decodifica codarea lui Peter și va realcătuși topologia stației.

Să se realizeze un program care să rezolve problema.

Date de intrare

Fișierul de intrare **recon.in** conține pe prima linie un număr N de module. Urmează $N - 1$ linii. Pe prima linie se dau distanțele de la modulul 1 la modulele 2, 3, ..., N , separate de spații. Pe a doua linie se dau distanțele de la modulul 2 la modulele 3, 4, ..., N și așa mai departe. Ultima linie conține doar distanța de la modulul $N - 1$ la modulul N .

Date de ieșire

Programul trebuie să scrie N linii în fișierul de ieșire **recon.out**. Prima linie va conține lista vecinilor modulului 1, adică modulele legate prin coridoare de el. Lista trebuie să înceapă cu numărul L de vecini, urmat de numerele lor de ordine, ordonate crescător. Toate numerele vor fi separate printr-o singură spațiu. Pe a doua linie se va scrie, formatată în același mod, lista vecinilor modulului 2 și așa mai departe. Fișierul se va termina cu lista vecinilor modulului N .

Restricții

$$3 \leq N \leq 1024$$

Distanțele $D_{i,j}$ sunt întregi pozitivi mai mici strict decât 15000.

10. Ninjago¹⁰

Zane trebuia să păzească cele n păpuși dintr-un muzeu. Între aceste păpuși există m coridoare pe care se poate circula în ambele sensuri. Se garantează faptul că pe cele m coridoare Zane poate ajunge la fiecare dintre cele n păpuși.

Skulkiu, având la dispoziție 5 tipuri de obstacole (A , B , C , D , E), încearcă să-l oprească pe Zane punând pe fiecare coridor câte 4 obstacole. Zane poate distruge obstacolele de tip A , B , C și D , dar nu poate să distrugă obstacolele de tipul E .

¹⁰ <https://www.pbinfo.ro/probleme/2127/ninjago>

Pentru a distruge un obstacol de tipul A arma lui Zane are nevoie de 1 unitate de energie, pentru a distruge un obstacol de tipul B de 2 unități de energie, pentru a distruge un obstacol de tipul C de 3 unități de energie, iar pentru a distruge un obstacol de tipul D de 4 unități de energie. Datorită dispozitivului cu care Skulkiu amplasează obstacolele pe coridor, cele patru obstacole de pe același coridor au o adâncime din ce în ce mai mare, ceea ce implică faptul că pentru a distruge al doilea obstacol amplasat pe coridor este nevoie de 5 ori mai multă energie decât cea obișnuită, pentru a distruge cel de-al treilea obstacol amplasat pe coridor este nevoie de 25 ori mai multă energie decât cea obișnuită, iar pentru a distruge al patrulea obstacol amplasat pe același coridor este nevoie de 125 de ori mai multă energie decât cea obișnuită.

Indiferent de sensul de parcurgere al coridorului de către Zane pentru a înlătura obstacolele, energia consumată este aceeași, aceasta depinzând doar de ordinea în care au fost amplasate obstacolele de către Skulkiu. Zane nu va înlătura obstacolele de pe toate coridoarele ci doar strictul necesar pentru a avea acces la fiecare păpușă.

Zane dorește să-i lase pe ceilalți ninja să se antreneze așa că face în așa fel încât ajutorul pentru distrugerea obstacolelor de tip E să fie minim și apoi ca el să utilizeze un număr minim de unități de energie. Pentru coridoarele pe care se află obstacole de tip E Zane consumă energie doar pentru obstacolele de tip A, B, C și D. Inițial Zane se află lângă păpușa 1.

Răspundeți la următoarele cerințe:

1. Precizați la câte dintre cele n păpuși poate ajunge Zane înainte de a cere ajutorul celorlalți ninja.
2. Precizați pentru eliberarea câtor coridoare trebuie să ceară ajutor extern pentru a reuși să ajungă la toate cele n păpuși și câte obstacole de tip E sunt în total pe aceste coridoare.
3. Precizați care este numărul minim de unități de energie utilizate.

Date de intrare

Fișierul de intrare `ninjago.in` conține pe prima linie v care poate avea doar valorile 1, 2 sau 3 reprezentând cerința care va fi rezolvată. Fișierul `ninjago.in` conține pe a doua linie numerele naturale n și m separate printr-un spațiu, iar pe următoarele m linii pentru fiecare coridor două numere naturale separate printr-un spațiu reprezentând cele două păpuși între care se circulă pe coridorul respectiv urmate de un spațiu și patru litere corespunzătoare celor patru tipuri de obstacole în ordinea în care Skulkiu le-a amplasat pe coridorul respectiv. Între cele patru litere nu se află nici un spațiu.

Date de ieșire

Dacă valoarea lui v este 1 atunci fișierul de ieșire `ninjago.out` va conține pe prima linie numai numărul păpușilor la care poate ajunge Zane înainte de a cere ajutorul celorlalți ninja.

Dacă valoarea lui v este 2 atunci fișierul de ieșire `ninjago.out` conține pe prima linie numărul minim de coridoare pentru eliberarea cărora Zane trebuie să ceară ajutor extern, pentru a reuși să ajungă la toate cele N păpuși, iar pe a doua linie numărul minim de obstacole de tip E ce se află în total pe o astfel de mulțime de coridoare.

Algoritmica grafurilor

Dacă valoarea lui v este 3 atunci fișierul de ieșire `ninjago.out` va conține pe prima linie doar numărul minim de unități de energie utilizate, în contextul unei soluții de la întrebarea de mai sus.

Restricții

$$1 \leq N, M \leq 31200 .$$