

Problema 1 - Comunităţi

100 de puncte

Elful Bushy Evergreen lucrează, de ceva vreme, la reţeaua de socializare **earsbook** pentru elfii din Elfland. A început prin a le crea tuturor elfilor câte un cont pe acest site şi a le da posibilitatea să se împrietenească. Evident, nu toţi elfii vor face acest lucru, prin urmare este posibil ca, în final, să existe elfi care, deşi au cont, nu au niciun prieten pe **earsbook**. Rând pe rând, elfii încep să se împrietenească şi să formeze **comunităţi**. Prin **comunitate** înţelegem o mulţime *cu cel puţin doi elfi* în care, din aproape în aproape, se poate găsi un lanţ de prieteni, oricât de întortocheat, între oricare doi elfi ai mulţimii. Spre exemplu, dacă avem următoarele relaţii de prietenie: (1,2) (2,3) (3,4) (4,5) (3,6) putem spune despre mulţimea {1,2,3,4,5,6} că formează o comunitate.

Date fiind **n** - numărul de elfi şi **m** - numărul de relaţii de prietenie, iar apoi cele **m** relaţii de prietenie, în ordinea creării lor, ajutaţi-l pe Bushy Evergreen să determine atât numărul final de comunităţi cât şi numărul maxim de comunităţi existente la un moment dat, pe parcursul creării relaţiilor de prietenie, precum şi lungimea secvenţei maxime de relaţii pe parcursul căreia numărul de comunităţi a stagnat (nici nu a crescut, nici nu a scăzut).

Date de intrare:

Programul citeşte, de pe prima linie a fişierului **comunitati.in**, numerele **n** şi **m**, iar de pe fiecare dintre următoarele **m** linii câte două valori **x** şi **y** separate prin spaţiu, reprezentând o relaţie de prietenie;

Date de ieşire:

Programul afişează în fişierul **comunitati.out** cele 3 valori cerute:

- **nfin** - numărul final de comunităţi,
- **nmax** - numărul maxim de comunităţi,
- **st** - lungimea secvenţei maxime de relaţii pe parcursul căreia numărul de comunităţi a stagnat.

Restricţii şi precizări:

Restricţii şi precizări

- $1 \leq n \leq 100000$
- $1 \leq m \leq 500000$
- o relaţie de prietenie (x,y) este reciprocă, iar în fişier va fi descrisă în mod unic, fie prin perechea x y, fie prin perechea y x.

Exemplu:

comunitati.in	comunitati.out	Explicații
11 13 4 6 1 6 2 5 3 7 10 11 5 3 1 4 2 7 6 7 1 5 2 4 8 9 9 10	2 4 3	4-6 formează prima comunitate ◯avem 1 comunitate 1 se adaugă la aceeași comunitate ◯tot 1 comunitate 2-5 formează a doua comunitate ◯avem 2 comunități 3-7 formează a treia comunitate ◯avem 3 comunități 10-11 formează a treia comunitate ◯avem 4 comunități 5-3 reunește comunitățile 2 și 3 ◯revenim la 3 comunități 1-4 se unesc noduri care sunt deja în comunitatea 1 ◯avem 3 comunități 2-7 se unesc noduri care sunt deja în comunitatea 1 ◯avem 3 comunități 6-7 reunește două comunități ◯revenim la 2 comunități 1-5 se unesc noduri care sunt deja în aceeași comunitate ◯rămânem la 2 comunități 2-4 se unesc noduri care sunt deja în aceeași comunitate ◯rămânem la 2 comunități 8-9 formează o nouă comunitate ◯avem 3 comunități 9-10 reunește două comunități ◯revenim la 2 comunități

Obs: Lungimea secvenței celei mai lungi pe parcursul căreia numărul de comunități rămâne același este 3 (sunt două astfel de secvențe: (5,3) (1,4) (2,7) respectiv (6,7) (1,5) (2,4)).

Timp maxim de execuție: 0,4 secunde/test

Memorie totală disponibilă: 1 MB

Dimensiunea maximă a sursei: 15 KB