One-Way Streets

Time Limit: 3 s *Memory Limit: 256 MB*

A fost odata o tara cu N orase si M sosele bidirectionale care le conecteaza. Dezvoltarea tenologica a dus la masini mai mari si mai rapide, ceea ce a reprezentat o problema — soselele au devenit prea inguste pentru vehiculele care merg in directii opuse. O decizie pentru rezolvarea problemei ar insemna transformarea soselelor in drumuri cu sens unic (unidirectionale).

A face soselele unidirectionale vine cu un dezavantaj deoarece unele perechi de orase care erau anterior accesibile e posibil ca acum sa nu mai fie. Guvernul a construit o lista de perechi importante de orase intre care trebuie sa existe posibilitatea de a porni de la primul oras si sa se ajunga la al doilea. Sarcina voastra este sa determinati in ce directie trebuie stabilit sensul pe fiecare sosea. Se garanteaza ca exista o astfel de solutie.

Pentru anumite sosele nu exista mai multe posibilitati legat de sensul traficului daca vrei sa obtii o solutie. Traficul va decurge de la primul catre al doilea oras (directia spre dreapta, indicata de litera `R`) sau de la al doilea oras catre primul (directia stanga, indicata de litera `L`). Totusi pentru unele drumuri poate sa existe o solutie cu aceasta sosea directionata spre stanga, iar alta solutie cu soseaua directionata spre dreapta. Trebuie indicate aceste sosele cu litera `B` pentru ambele directii.

Afisati un sir de caractere de lungime M. Al i-lea caracter ar trebui sa fie

- `R` daca toate solutiile necesita a i-a sosea sa fie directionata spre dreapta
- `L` daca toate solutiile necesita a i-a sosea sa fie directionata spre stanga
- `B` daca exista o solutie in care a i-a sosea este directionata spre stanga, iar o alta solutie exista cu a i-a sosea directionata spre dreapta

Intrare

Prima linie contine numarul de orase N si numarul de sosele M. Urmatoarele M linii descriu soselele prin perechi de numere a_i si b_i, ce indica faptul ca exista o sosea intre orasele a_i si b_i. Poate exista mai mult de o sosea intre aceeasi pereche de orase iar un drum chiar poate conecta un oras cu el insusi.

Urmatoarea linie contine numarul P de perechi de orase pentru care trebuie sa ramana posibilitatea de a ajunge de la unul la celalalt. Urmatoarele P linii contin perechi de orase x_i si y_i, insemnand ca trebuie sa existe o modalitate de a porni din orasul x_i si sa se ajunga in y_i.

Restrictii

- $1 \leq N, M, P \leq 100\,000$
- $1 \leq a_i, b_i, x_i, y_i \leq N$

Subtask 1 (30 de puncte)

- $N, M \leq 1000$
- $P \leq 100$

Subtask 2 (30 de puncte)

- $P \leq 100$

Subtask 3 (40 de puncte)

- fara restrictii

Output

Afisati un sir de caractere de lungime M asa cum s-a explicat in enunt.

Exemplu

Input	Output
5 6 1 2 1 2 4 3 2 3 1 3 5 1 2 4 5 1 3	BBRBBL

Explicatii

Sa aratam ca a cincea sosea "1 3" poate fi orientata in ambele directii. Doua posibile orientari ale soselelor cu directii diferite pentru a cincea sosea sunt `LLRLRL` si `RLRRLL`.

Sure Bet

Time Limit: 2 s *Memory Limit: 128 MB*

Bulanul este o parte fundamentala a pariurilor. Unii oameni pot sa isi imbunatateasca sansele de castig daca au o buna cunoastere a pariurilor pe care le fac. Noi vom aborda diferit.

Casele de pariuri pot oferi diferite *cote* pentru acelasi rezultat. (O *cota* de x inseamna ca daca pariezi 1 euro si prezici rezultatul cum trebuie, castigi x euro inapoi. Daca prezici rezultatul gresit, nu primesti nimic inapoi. De remarcat ca platesti 1 euro indiferent de verdictul pariului. Cum ar fi daca ai putea sa fii sigur de un anumit profit daca ai paria in mod inteligent? Ti-ai dori ca acest profit garantat sa fie cat mai mare cu putinta.

Evenimentul la care vom participa are doua posibile rezultate. Avem N case de pariuri care vor oferi diferite cote. Sa notam cota oferita de casa de pariuri cu indicele i pentru primul rezultat cu a_i, si cota oferita pentru cel de al doilea rezultat cu b_i. Se poate paria pe orice subset de cote oferite, inclusiv pe ambele rezultate la aceasi casa de pariuri. Cu toate acestea, toate pariurile trebuie sa fie de 1 euro exact si nu poti paria de mai multe ori acelasi rezultat la aceasi casa de pariuri.

In cazul primului verdict, vei primi a_i euro de la fiecare casa de pariuri i la care ai pariat pe primul rezultat. Similar, in cazul celui de al doilea verdict, vei primi b_i euro de la toate casele de pariuri eligibile. Desigur, in ambele cazuri deja ai platit 1 euro pentru fiecare pariu facut.

Care este cel mai mare profit *garantat* (i.e. indiferent de rezultat) pe care il poti obtine daca pariezi optim?

Input

Pe prima linie se afla numarul caselor de pariuri, N. Urmatoarele N linii descriu cotele oferite de fiecare casa de pariuri prin doua numere reale a_i si b_i (cota primului respectiv celui de al doilea verdict dat de casa de pariuri cu indicele i), separate prin cate un spatiu. Cotele vor fi date cu cel mult 4 zecimale.

Restrictii

- $1.0 \leq a_i, b_i \leq 1000.0$
- $1 \leq N \leq 100\,000$

Subtask 1 (20 de puncte)

- $N \leq 10$

Subtask 2 (40 de puncte)

- $N \leq 1\,000$

Subtask 3 (40 de puncte)

- fara restrictii suplimentare

Output

Afisati profitul maxim garantat pe care il puteti obtine, rotunjit cu exact 4 zecimale.
Acestea sunt comenzile de afisare a numerelor reale in diferite limbaje:

- C si C++: `printf("%.4lf",(double)x);`
- Java: `System.out.printf("%.4lf",x);`
- Pascal: `writeln(x:0:4);`
- Python 3: `print("%.4lf"%x)`
- C#: `Console.WriteLine(String.Format("0:0.0000",x));`

Example

Input	Output
4 1.4 3.7 1.2 2 1.6 1.4 1.9 1.5	0.5000

Explicatii

Strategia optima de castig presupune in a paria pe al doilea rezultat la prima casa de pariuri si primul rezultat la cea de a treia si a patra casa de pariuri. In cazul primului verdict, vom castiga $1.6 + 1.9 - 3 = 0.5$ iar in cazul celui de al doilea verdict $3.7 - 3 = 0.7$. Deci este garantat sa castigam 0.5 euro, indiferent de rezultat.

Mousetrap

Time Limit: 5 s *Memory Limit: 512 MB*

Dumbo elefantul are un labirint gigantic cu N camere numerotate de la 1 la N si $N-1$ pasaje in asa fel incat este posibil sa ajungi din orice camera in orice alta camera. Din pacate, un soarece s-a strecurat in labirint. Dumbo este speriat de moarte de soareci, asa ca a pus o capcana in camera T. Desigur, soarecele evita camera cu capcana, deci Dumbo trebuie sa se gandeasca la o strategie mai buna pentru a il ademeni in capcana. Soarecele alearga in mod constant si niciodata nu se opreste, decat daca nu are unde sa se duca. De asemenea, el stie ca soarecele lasa o urma murdara prin fiecare pasaj prin care trece iar acesta refuza sa foloseasca un pasaj murdar din nou. Dumbo poate sa curete un pasaj sau sa il blocheze cu pietre. Prin blocarea sau curatarea pasajelor, acesta doreste sa forteze soarecele sa ajunga la capcana. Ar dori sa faca asta intr-un numar minim de mutari, din moment ce se simte incomod in prezenta sobolanului.

Putem descrie acest macel ca un joc intre doi jucatori. Soarecele incearca sa maximizeze numarul de mutari a lui Dumbo, iar Dumbo incearca sa castige intr-un numar minim de operatii. In tura sa, acesta poate sa curete un pasaj murdar a labirintului sau sa blocheze orice pasaj. Nu conteaza daca pasajul blocat este curat sau nu. De asemenea, un pasaj nu poate sa fie deblocat. In schimb, acesta poate sa aleaga sa nu faca nimic. Turele in care Dumbo decide sa nu faca nimic nu se numara la solutie. Cand vine randul soarecelui, acesta va alege un pasaj curat neblocat si va alerga in camera vecina catre care pasajul indica. Daca nu exista un astfel de pasaj ce porneste din camera curenta, soricelului va sta pe loc.

Initial, toate pasajele sunt curate, soarecele se afla in camera M, camera cu capcana este T, iar Dumbo incepe. Care este numarul minim de mutari (curatari de pasaje si blocari) de care Dumbo are nevoie daca ambii jucatori joaca optim (soricelul incearca sa maximizeze numarul de mutari a lui Dumbo)?

Input

Pe prima linie se afla intregii N, T si M, separate prin cate un spatiu. Pe urmatoarele $N-1$ linii vor fi cate doua numere naturale a_i si b_i, separate prin cate un spatiu, indicand ca exista un pasaj intre camerele a_i si b_i.

Atentie la marimea inputului deoarece este foarte mare.

Restrictii

- $1 \leq N, T, M \leq 10^6$

Subtask 1 (20 de puncte)

- $N \leq 10$

Subtask 2 (25 de puncte)

- Se garanteaza ca exista pasaj intre camerele M si T.

Subtask 3 (20 de puncte)

- $N \leq 1000$

Subtask 4 (35 de puncte)

- fara restrictii suplimentare

Output

Programul trebuie sa afiseze numarul de mutari a lui Dumbo.

Example

Input	Output
10 1 4 1 2 2 3 2 4 3 9 3 5 4 7 4 6 6 8 7 10	4

Explicatii

Un posibil scenariu:

- Dumbo blocheaza pasajul dintre camerele 4 si 7.
- Soarecele se muta in camera 6. Pasajul dintre camere 4 si 6 este acum murdar.
- Dumbo blocheaza pasajul dintre camerele 6 si 8.
- Soarecele nu poate muta.
- Dumbo curata pasajul dintre camerele 4 si 6.
- Soarecele se muta in camera 4. Pasajul dintre camerele 4 si 6 este din nou murdar.
- Dumbo blocheaza pasajul dintre camerele 2 si 3.
- Soarecele se muta in camera 2. Pasajul dintre camerele 2 si 4 este murdar.
- Dumbo nu face nimic.
- Soarecele se poate muta doar in camera 1 si isi ia tzeapa in capcana.

Dumbo a facut 4 mutari.