One-Way Streets

Vremensko ograničenje: 3 s *Memorijsko ograničenje: 256 MB*

Jednom davno, ali ne tako davno, postojala je država u kojoj je bilo n gradova i m dvosmjernih cesta koje su ih povezivale. Kako je ta država rasla i postojala sve uglednija i naprednija, konstruirala je sve veća i veća vozila za putovanje tim cestama. Međutim, napredak je zaprijetio njenom opstanku jer se pojavio problem: dva vozila se više nisu mogla mimoići jer su ceste postale preuske. Mudre vođe te zemlje donijele su jedinu logičnu odluku! Sve će ceste, koje su sada dvosmjerne, pretvoriti u jednosmjerne.

Na žalost, zbog ovog rješenja neki od parova gradova koji su prije bili povezani nakon promjene više neće biti dostupni. Zato su voljene vođe te zemlje napravile popis parova gradova za koje će nakon promjene uvijek biti moguće krenuti iz prvog grada u paru i doći do drugog grada u paru. Tvoj zadatak je za svaku cestu odrediti u kojem smjeru će se promet kretati. Garantira ti se da rješenje postoji.

Za neke ceste nećemo imati mogućnost izbora smjera ako želimo dobiti rješenje. Promet će teći od prvog do drugog grada (desni smjer, slovna oznaka `R`) ili od drugog do prvog grada (lijevi smjer, slovna oznaka `L`). Međutim, za neke ceste može vrijediti da postoji rješenje u kojemu je ta cesta lijevo usmjerena, a postoji i drugo rješenje (po mogućnosti različito) u kojemu je ta cesta desno usmjerena. Takve ceste trebaš označiti slovnom oznakom `B`.

Izlaz je string duljine m. Njegov i-ti znak je

- `R` ako je u svim rješenjim i-ta cesta desno usmjerena
- `L` ako je u svim rješenjim i-ta cesta lijevo usmjerena
- `B` ako postoji rješenje u kojem je i-ta cesta lijevo usmjerena, a postoji i rješenje u kojem je i-ta cesta desno usmjerena

Ulazni podaci

U prvom retku nalazi se broj gradova n i broj cesta m. U sljedećih m redaka nalaze se parovi brojeva a_i i b_i, koji označavaju da postoji cesta između gradova a_i i b_i. Moguće je da postoji više cesta između dva ista para gradova, a moguće je i da postoji cesta koja povezuje grad sa samim sobom.

Sljedeća linija sadrži broj parova gradova p iz teksta zadatka. Sljedećih p linija ulaza sadrže po par gradova x_i i y_i, za koje će morati postojati put koji počinje u gradu x_i i završava u gradu y_i.

Ograničenja

- $1 \leq n, m, p \leq 100\,000$
- $1 \leq a_i, b_i, x_i, y_i \leq n$

Podzadatak 1 (30 bodova)

- $n, m \leq 1000$
- $p \leq 100$

Podzadatak 2 (30 bodova)

- $p \leq 100$

Podzadatak 3 (40 bodova)

- nema dodatnih ograničenja

Izlazni podaci

Izlaz je string iz teksta zadatka.

Primjer test podatka

Ulaz	Izlaz
5 6 1 2 1 2 4 3 2 3 1 3 5 1 2 4 5 1 3	BBRBBL

Opis prvog test podatka

Primjeti da se peta cesta "1 3" može lijevo usmjeriti, a može se i desno usmjeriti. Dva moguća rješenja s različitim usmjerenjem za petu cestu su: `LLRLRL` and `RLRRLL`.

Mousetrap

Vremensko ograničenje: 5 s *Memorijsko ograničenje: 512 MB*

Slon Dumpo posjeduje ogroman labirint s n soba označenih brojevima $1 \dots n$ i $n-1$ hodnika postavljenih tako da postoji put između svake dvije sobe. Nažalost, miš se ušuljao u labirint. Dumpo se užasava miševa stoga je postavio mišolovku u sobu broj t. Miš očigledno želi izbjeći sobu sa zamkom, pa Dumpo mora osmisliti bolju strategiju da bi navabio miša u zamku. Miš neprestano trčkara okolo te se nikad ne zaustavlja, osim ako se ne može nigdje pomaknuti. Dumpo također zna da miš ostavlja prljavi trag izmeta i otisaka u svakom hodniku kroz koji prođe. Miš odbija proći kroz prljavi hodnik. Dumpo može očistiti prljavi hodnik ili blokirati neki hodnik kamenjem. Čisteći i blokirajući hodnike, želi natjerati miša da uleti u zamku. To želi postići u najmanjem mogućem broju poteza jer se osjeća vrlo nelagodno u blizini miša.

Ovo možemo opisati kao igru za dva igrača. Mišev cilj je maksimizirati broj Dumpovih poteza, dok on pokušava pobijediti u što manjem broju poteza. Dumpo igra prvi. Kada je na potezu, može očistiti prljavi hodnik labirinita ili blokirati hodnik bez obzira je li on prljav ili nije. Ne može odblokirati hodnik, međutim može odlučiti ne učiniti ništa. Ako nije ništa učinio, to se ne broji kao potez. Kada je miš na redu, odabrat će čisti hodnik i otići kroz njega u susjednu sobu. Ako takav hodnik ne postoji, miš će ostati na mjestu.

Na početku, svi hodnici su čisti, miš se nalazi u sobi s brojem m, zamka u sobi s brojem t, i Dumpo je na potezu. Koji je minimalan broj poteza (čišćenja i blokiranja hodnika) koje Dumpo treba napraviti ako oba igrača igraju optimalno (mišev clj je maksimizirati broj Dumpovih poteza)?

Ulazni podaci

U prvom retku dani su cijeli brojevi n, t i m, odvojeni razmacima. U sljedećih $n-1$ redaka nalaze se brojevi a_i i b_i, odvojeni razmakom, koji predstavljaju hodnik između soba a_i i b_i.

Primijetite da je veličina ulaza velika.

Ograničenja

- $1 \leq n, t, m \leq 10^6$

Podzadatak 1 (20 bodova)

- $n \leq 10$

Podzadatak 2 (25 bodova)

- Garantirano je da postoji hodnik između soba m i t.

Podzadatak 3 (20 bodova)

- $n \leq 1000$

Podzadatak 4 (35 bodova)

- bez dodatnih ograničenja

Izlazni podaci

Tvoj program mora ispisati broj Dumpovih poteza.

Primjer test podatka

Ulaz	Izlaz
10 1 4 1 2 2 3 2 4 3 9 3 5 4 7 4 6 6 8 7 10	4

Komentar

Jedan mogući scenarij:

- Dumpo blokira hodnik između soba 4 i 7.
- Miš se pomakne u sobu 6. Hodnik između soba 4 i 6 je sada prljav.
- Dumpo blokira hodnik između soba 6 i 8.
- Miš se ne može pomaknuti.
- Dumpo čisti hodnik između soba 4 i 6.
- Miš se pomakne u sobu 4. Hodnik između soba 4 i 6 je prljav.
- Dumpo blokira hodnik između soba 2 i 3.
- Miš se pomakne u sobu 2. Hodnik između soba 2 i 4 je prljav.
- Dumpo ne čini ništa.
- Miš se jedino može pomaknuti u sobu broj 1 i biti uhvaćen u zamku.

Dumpo je učinio 4 poteza.

Sure Bet

Vremensko ograničenje: 2 s *Memorijsko ograničenje: 128 MB*

Opće je poznato kako je sreća najvažniji aspekt klađenja. Usprkos tome, ljudi poput Kileta i Pogija koriste razna sportska znanja kako bi povećali svoje šanse i popunili optimalan tiket. Mi nismo poput Kileta i Pogija pa ćemo svoje šanse povećati na potpuno drugačiji način.

Naime, razne kladionice (u daljnjem tekstu kladare) nude razne *koeficijente* za iste ishode utakmica. (*Koeficijent* x na neki ishod znači da ćete, uplatite li 1 euro na taj ishod, od kladare dobiti točno x eura natrag. Naravno, ako ste pogriješili u procjeni ishoda događaja, nećete osvojiti ništa). Zamislite svijet u kojem je moguće lukavo odraditi neke oklade tako da vam je profit zagarantiran. Za potrebe ovog zadatka živite upravo u takvom svijetu i želite maksimizirati vaš zagarantirani profit.

Događaj na koji se želimo kladiti ima dva moguća ishoda, a u našem se gradu nalazi n kladara koje nude razne koeficijente za te ishode. Formalnije, koeficijent koji i-ta kladara nudi za prvi ishod označavamo sa a_i, a koeficijent koji ta ista kladara nudi za drugi ishod označavamo sa b_i. Možete se kladiti na bilo koji podskup ponuđenih koeficijenata, dakle, možete se kladiti čak i na oba ishoda koje nudi ista kladara. S druge strane, vaš ulog na svaku od oklada mora biti točno 1 euro te ne smijete napraviti više od jedne oklade na neki ishod unutar iste kladare.

U slučaju prvog ishoda, osvojit ćete a_i eura od svake kladare i u kojoj ste se kladili na prvi ishod. Analogno, u slučaju drugog ishoda, osvojit ćete b_i eura iz svake kladare i u kojoj ste se kladili na drugi ishod. Naravno, u svakom slučaju već ste platili 1 euro koji ste uložili u okladu.

Koji je najveći *zagarantirani* profit (neovisno o ishodu) ako ste se optimalno kladili?

Ulazni podaci

U prvom retku nalazi se broj kladara, n. Svaki od sljedećih n redaka sadrži koeficijente koje nude kladare. Preciznije, u $i+1$-om retku nalaze se realni brojevi a_i i b_i koji redom označavaju koeficijent za prvi, odnosno drugi ishod koji nudi i-ta kladara. Koeficijenti neće sadržavati više od 4 znamenke nakon decimalne točke.

Ograničenja

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

Podzadatak 1 (20 bodova)

- $n \leq 10$

Podzadatak 2 (40 bodova)

- $n \leq 1\,000$

Podzadatak 3 (40 bodova)

- nema dodatnih ograničenja

Izlazni podaci

U prvi i jedini redak ispišite najveći mogući zagarantirani profit zaokružen na točno 4 decimalna mjesta.

Slijedi kratak podsjetnik na uobičajen način ispisa brojeva sa posmičnim zarezom u dopuštenim programskim jezicima:

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

Primjeri test podataka

Ulaz	Izlaz
4 1.4 3.7 1.2 2 1.6 1.4 1.9 1.5	0.5000

Komentar

Optimalno klađenje sastoji se od oklada na drugi ishod u prvoj kladari te oklada na prvi ishod u trećoj i četvrtoj kladari. U slučaju da se dogodi prvi ishod, zaradit ćete $1.6 + 1.9 - 3 = 0.5$, a u slučaju da se dogodi drugi ishod zaradit ćete $3.7 - 3 = 0.7$. Dakle, neovisno o ishodu zagarantirana nam je zarada od 0.5 eura.