



Skierowanie dróg

Limit czasu: 3 s Limit pamięci: 256 MB

W pewnym państwie jest n miast i m dróg, do tej pory dwukierunkowych. Dzięki postępowi technologicznemu pojazdy stały się szybsze i większe niż kiedyś. Doprowadziło to niestety do problemu – drogi stały się zbyt wąskie, by dwa pojazdy wyminęły się, jadąc w przeciwnych kierunkach. Podjęto zatem decyzję, by skierować każdą drogę, czyli uczynić wszystkie drogi jednokierunkowymi.

Z sobie tylko znanych powodów, rząd państwa przygotował listę p uporządkowanych par miast (x_i, y_i) . Konieczne jest, by dla każdej takiej pary dało się dostać z miasta x_i do miasta y_i (skierowanie dróg, spełniające ten warunek, nazwijmy poprawnym). Rząd nie stawia oczywiście warunków niemożliwych do spełnienia – możesz założyć, że istnieje co najmniej jedno poprawne skierowanie dróg.

Mówimy, że droga (a_i, b_i) może być skierowana w prawo, jeśli istnieje poprawne skierowanie dróg, w którym i -ta droga jest skierowana z a_i do b_i . Podobnie powiemy, że droga może być skierowana w lewo, gdy w pewnym poprawnym skierowaniu dróg jest ona skierowana z b_i do a_i . Dla każdej z m dróg musisz stwierdzić, czy może być skierowana w lewo, prawo, czy może w obu kierunkach.

Wypisz słowo o długości m , którego i -ty znak to:

- R jeśli i -ta droga może być skierowana tylko w prawo,
- L jeśli i -ta droga może być skierowana tylko w lewo,
- B jeśli i -ta droga może być skierowana w prawo oraz może być skierowana w lewo.

Wejście

Pierwszy wiersz wejścia zawiera liczbę miast n oraz liczbę dróg m . i -ty z kolejnych m wierszy zawiera dwie liczby a_i i b_i , opisujące dwukierunkową drogą między miastami a_i oraz b_i .

Dwa miasta mogą być połączone więcej niż jedną drogą. Droga może łączyć miasto z samym sobą.

Kolejny wiersz zawiera liczbę par miast p . i -ty z kolejnych p wierszy zawiera dwie liczby x_i i y_i , oznaczające że każde poprawne skierowanie musi pozwalać na dostanie się z miasta x_i do miasta y_i .

Możesz założyć, że istnieje co najmniej jedno poprawne skierowanie dróg.

Ograniczenia

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

Podzadanie 1 (30 punktów)

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



Podzadanie 2 (30 punktów)

- $p \leq 100$

Podzadanie 3 (40 punktów)

- brak dodatkowych ograniczeń

Wyjście

Wypisz słowo o długości m , jak opisano powyżej.

Przykład

Wejście

5 6
1 2
1 2
4 3
2 3
1 3
5 1
2
4 5
1 3

Wyjście

BBRBBL

Komentarz

Pokażmy, że piąta droga „1 3” może być skierowana w lewo i może być skierowana w prawo (czyli piąta literka wyjścia to 'B'). Możesz sprawdzić, że dwa poprawne skierowania dróg to LLRLRL oraz RLRLL. W pierwszym z nich piąta droga jest skierowana w prawo (oznaczone literką 'R'), a w drugim skierowaniu ta droga jest skierowana w lewo ('L').



Pewny zakład

Limit czasu: 2 s Limit pamięci: 128 MB

Szczęście jest podstawą zakładów bukmacherskich. Część ludzi zwiększa swoje szanse (a tym samym zyski) poprzez dobrą znajomość tego, co obstawiają. My jednak spróbujemy innego pomysłu.

Różni bukmacherzy oferują różne kursy dla tych samych rezultatów tego samego wydarzenia. (*Kurs* równy x oznacza, że jeśli postawisz złotówkę i odgadniesz poprawnie, otrzymasz x złotych wygranej. W przeciwnym przypadku oczywiście nie otrzymujesz żadnej wygranej. Zawsze jednak za zakład płacisz złotówkę niezależnie od rezultatu.) Co gdybyś mógł być pewnym zysku poprzez mądre obstawianie u wielu bukmacherów? Chcesz uzyskać maksymalny pewny zysk.

Wydarzenie, którego rezultat chcemy obstawiać, ma dwa możliwe rezultaty. Jest n bukmacherów, którzy oferują różne kursy. Kurs na pierwszy rezultat oferowany przez i -tego bukmachera oznaczamy jako a_i , zaś kurs tego bukmachera na drugi rezultat oznaczamy jako b_i . Możesz obstawić dowolny podzbiór rezultatów u dowolnie wybranych bukmacherów. Dozwolone jest nawet, aby obstawić oba rezultaty u tego samego bukmachera. Każdy zakład jest za złotówkę i nie możesz obstawiać tego samego rezultatu u tego samego bukmachera wielokrotnie.

W przypadku pierwszego rezultatu otrzymasz a_i złotych od każdego bukmachera i , u którego postawiłeś na pierwszy rezultat. Podobnie, w przypadku drugiego rezultatu, otrzymasz b_i złotych od wszystkich bukmacherów, u których ten rezultat obstawiłeś. Oczywiście w obu przypadkach płacisz po złotówce za każdy obstawiony rezultat.

Jaki jest największy *pewny* zysk (tzn. niezależnie od rezultatu obstawianego wydarzenia) jeśli obstawisz optymalnie?

Wejście

Pierwszy wiersz wejścia zawiera liczbę bukmacherów n . Kolejne n wierszy opisuje kursy oferowane przez każdego bukmachera – dwie liczby rzeczywiste a_i oraz b_i , oddzielone pojedynczym odstępem, opisujące kursy na pierwszy oraz drugi rezultat u i -tego bukmachera. Kursy będą podane z dokładnością do co najwyżej czterech cyfr po kropce dziesiętnej.

Ograniczenia

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

Podzadanie 1 (20 punktów)

- $n \leq 10$

Podzadanie 2 (40 punktów)

- $n \leq 1\,000$

Podzadanie 3 (40 punktów)

- brak dodatkowych ograniczeń



Wyjście

Wypisz maksymalny pewny zysk z dokładnością do dokładnie czterech cyfr po kropce dziesiętnej.

Poniżej znajdują się instrukcje do wypisania liczb zmiennoprzecinkowych w różnych językach programowania:

- C i 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));`

Przykład

Wejście

4
1.4 3.7
1.2 2
1.6 1.4
1.9 1.5

Wyjście

0.5000

Komentarz

Optymalna strategia polega na obstawieniu drugiego rezultatu u pierwszego bukmachera oraz pierwszego rezultatu u trzeciego i czwartego bukmachera. W przypadku pierwszego rezultatu wydarzenia zarobisz $1.6 + 1.9 - 3 = 0.5$ złotych, a w przypadku drugiego rezultatu $3.7 - 3 = 0.7$ zł. Gwarantowany zysk wynosi więc 0.5 zł.



Pułapka Mariusza

Limit czasu: 5 s Limit pamięci: 512 MB

Słoń Mariusz czuwa nad dużym labiryntem, złożonym z n pokoiów ponumerowanych kolejnymi liczbami naturalnymi od 1 do n , połączonych $n - 1$ korytarzami. Z każdego pokoju da się dotrzeć do każdego innego pokoju.

Niestety, do labiryntu wślizgnęła się mysz. Słoń Mariusz czuje się wysoce niekomfortowo w obecności małych zwierzątek, więc przyrzucił pułapkę na myszy w pokoju t , do którego trzeba zwabić małego szkodnika. Mysz jest ruchliwa i zawsze przemieszcza się do jednego z sąsiednich pokoiów, chyba że nie ma takiej możliwości. Wiadomo też, że mysz zostawia za sobą ślady i okruszki, co czyni użyty korytarz brudnym. Mariusz jest w stanie wyczyścić brudny korytarz lub całkowicie zablokować jakikolwiek (brudny lub czysty) korytarz kamieniami. Wybredna mysz używa tylko czystych niezablokowanych korytarzy. Mariusz chce zmusić mysz do wejścia do pokoju z pułapką, używając do tego jak najmniejszej liczby ruchów (ruchem jest zablokowanie lub wyczyszczenie korytarza).

Możemy zdefiniować następującą grę dla dwóch graczy. Mysz próbuje zmaksymalizować liczbę ruchów Mariusza, a Mariusz chce tę liczbę zminimalizować. Mariusz zaczyna grę. W każdej jego turze, może on wyczyścić brudny korytarz, zablokować jakikolwiek korytarz lub nic nie robić. Nie może on odblokować zablokowanego korytarza. Tury, w których Mariusz nic nie robi, nie liczą się jako ruchy. W swojej turze mysz wybiera jeden z czystych niezablokowanych korytarzy wychodzących z jej obecnego pokoju i używa go, by przemieścić się do sąsiedniego pokoju. Jeśli nie ma żadnego takiego korytarza, mysz pozostaje w obecnym pokoju. Gra kończy się, gdy mysz wchodzi do pokoju z pułapką.

Początkowo, wszystkie korytarze są czyste i niezablokowane, mysz jest w pokoju m , a pułapka w pokoju t . Mariusz zaczyna grę. Jaka jest liczba ruchów Mariusza (liczba tur, gdy Mariusz czyści korytarz albo blokuje korytarz), jeśli Mariusz i mysz grają optymalnie? (Celem myszy jest zmaksymalizowanie liczby ruchów Mariusza.)

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby naturalne n , t oraz m , pooddzielane pojedynczymi odstępami. W kolejnych $n - 1$ wierszach znajduje się opis kolejnych korytarzy. W każdym wierszu znajdują się liczby naturalne a_i oraz b_i , oddzielone pojedynczym odstępem, oznaczające korytarz pomiędzy pokojami a_i oraz b_i .

Zauważ, że rozmiar wejścia jest duży.

Ograniczenia

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

Podzadanie 1 (20 punktów)

- $n \leq 10$

Podzadanie 2 (25 punktów)

- zawsze istnieje korytarz pomiędzy pokojami m oraz t .



Podzadanie 3 (20 punktów)

- $n \leq 1000$

Podzadanie 4 (35 punktów)

- brak dodatkowych ograniczeń

Wyjście

Twój program powinien wypisać na wyjście liczbę ruchów Mariusza.

Przykład

Wejście

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

Wyjście

4

Komentarz

Przykładowy możliwy scenariusz:

- Mariusz blokuje korytarz pomiędzy pokojami 4 i 7.
- Mysz przesuwa się do pokoju 6. Korytarz pomiędzy pokojami 4 i 6 staje się brudny.
- Mariusz blokuje korytarz pomiędzy pokojami 6 i 8.
- Mysz nie może się ruszyć.
- Mariusz czyści korytarz pomiędzy pokojami 4 i 6.
- Mysz przesuwa się do pokoju 4. Korytarz pomiędzy pokojami 4 i 6 staje się brudny.
- Mariusz blokuje korytarz pomiędzy pokojami 2 i 3.
- Mysz przesuwa się do pokoju 2. Korytarz pomiędzy pokojami 2 i 4 staje się brudny.
- Mariusz nic nie robi (bo jest leniwy).
- Mysz może się przesunąć tylko do pokoju 1 i zostać złapana w pułapkę Mariusza.

Mariusz wykonał cztery ruchy.