



# One-Way Streets

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

C'era una volta uno stato formato da  $n$  città ed  $m$  strade bidirezionali di collegamento. Lo sviluppo tecnico ha portato alla costruzione di veicoli più veloci e più ingombranti, che ha portato con sé un problema: le strade stavano diventando troppo strette per permettere a due veicoli provenienti da direzioni opposte di incrociarsi. È stato quindi deciso di risolvere il problema trasformando ogni strada in un senso unico (unidirezionale).

Purtroppo, rendere le strade a senso unico ha un costo, poichè alcune coppie di città che prima erano connesse potrebbero non essere più collegate dopo la trasformazione. Il governo ha formato una lista delle coppie notevoli di città per le quali è necessario poter partire dalla prima delle due e raggiungere la seconda. Il tuo compito è quello di determinare in quale direzione ridirigere il traffico per ogni strada. È garantito che esista almeno una soluzione.

Per alcune strade, non è possibile scegliere arbitrariamente la direzione del traffico per ottenere una soluzione: il traffico deve per forza scorrere dalla prima città alla seconda (direzione destra, indicata dalla lettera R) oppure dalla seconda alla prima (direzione sinistra, indicata dalla lettera L). Invece, per alcune strade esiste sia una soluzione in cui sono orientate verso destra che una (possibilmente diversa) in cui sono orientate verso sinistra. Queste strade vengono indicate con la lettera B, che significa “entrambe le direzioni”.

Devi stampare una stringa di lunghezza  $m$ . Il suo  $i$ -esimo carattere deve essere:

- R se tutte le soluzioni richiedono che l' $i$ -esima strada sia orientata verso destra
- L se tutte le soluzioni richiedono che l' $i$ -esima strada sia orientata verso sinistra
- B se esiste una soluzione dove l' $i$ -esima strada è orientata verso sinistra, ma anche una in cui l' $i$ -esima strada è orientata verso destra

## Limiti

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

### Subtask 1 (30 punti)

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

### Subtask 2 (30 punti)

- $p \leq 100$

### Subtask 3 (40 punti)

- nessuna limitazione aggiuntiva



## Input

La prima riga contiene il numero di città  $n$  e il numero di strade  $m$ . Le  $m$  righe successive descrivono le strade presenti, con una coppia di numeri  $a_i$  e  $b_i$  per l' $i$ -esima strada. Tali coppie indicano che c'è una strada tra la città  $a_i$  e la città  $b_i$ . Ci può essere più di una strada tra la stessa coppia di città e anche una strada che connette una città con se stessa.

La riga successiva contiene il numero di coppie di città  $p$  che devono essere raggiungibili. Le successive  $p$  righe contengono coppie di città  $x_i$  e  $y_i$ , col significato che ci deve essere un modo di partire dalla città  $x_i$  e arrivare nella città  $y_i$ .

## Output

Stampa una stringa di lunghezza  $m$  come spiegato nella descrizione del problema.

## Esempio

### Input

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

### Output

```
BBRBBL
```

## Nota

Mostriamo che la quinta strada "1 3" può essere orientata in entrambe le direzioni: LLRLRL e RLRRLL sono due possibili modi di orientare le strade tra le città in cui la quinta strada è orientata in modi diversi.



# Mousetrap

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

L'elefante Dumbo si trova in un grande labirinto con  $n$  stanze numerate  $(1 \dots n)$  e  $n - 1$  corridoi tra esse disposti in modo che sia sempre possibile raggiungere ogni stanza a partire da qualunque altra stanza. Sfortunatamente, un topolino si è infiltrato nel labirinto. Dumbo è terribilmente spaventato dai topi, quindi ha preparato una trappola nella stanza  $t$ . Ovviamente, il topolino evita la stanza con la trappola, quindi Dumbo deve pensare a una strategia migliore per attirare il topo nella trappola.

Il topo corre costantemente nel labirinto e non si ferma mai, a meno che non sia impossibile per lui muoversi. Dumbo sa anche che il topolino lascia una traccia di sporco e impronte dietro di sé in ogni corridoio che attraversa. Il topolino si rifiuta di attraversare un corridoio sporco. Dumbo può pulire un corridoio sporco o bloccare un corridoio usando delle pietre. Bloccando o pulendo i corridoi, Dumbo vuole forzare il topolino a correre verso la trappola. Dumbo tuttavia vorrebbe raggiungere questo obiettivo col numero minimo di mosse, poichè non si trova affatto a suo agio in presenza di un topo.

Possiamo descrivere questa situazione come un gioco a due giocatori. Il topolino cerca di massimizzare il numero di mosse di Dumbo, mentre Dumbo cerca di vincere nel minimo numero di mosse. Il primo giocatore è Dumbo: nel suo turno, può pulire un corridoio sporco oppure bloccare un qualsiasi corridoio. Il passaggio bloccato può essere sporco come no, ma Dumbo non può sbloccare un passaggio bloccato. Tuttavia, può scegliere di non fare niente in un certo turno e, in tal caso, tali turni non contano come mosse effettuate. Il topolino, durante il suo turno, sceglie un passaggio pulito e non bloccato e corre verso la stanza adiacente attraverso quel corridoio. Se un tale corridoio non esiste, il topolino non può muoversi.

Inizialmente tutti i corridoi sono puliti, il topolino si trova nella stanza  $m$ , la trappola nella stanza  $t$  e è il turno di Dumbo. Qual è il minimo numero di mosse (numero di volte che Dumbo ha pulito o bloccato un corridoio) che Dumbo deve fare se entrambi i giocatori giocano in maniera ottimale (l'obiettivo del topolino è massimizzare il numero di mosse di Dumbo)?

## Input

Gli interi  $n$ ,  $t$  e  $m$  sono scritti sulla prima riga, separati da spazi. Le successive  $n - 1$  righe contengono due interi  $a_i$  e  $b_i$ , separati da uno spazio, che indicano la presenza di un corridoio tra la stanza  $a_i$  e la stanza  $b_i$ .

Nota che l'input può essere molto grande.

## Limiti

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

### Subtask 1 (20 punti)

- $n \leq 10$

### Subtask 2 (25 punti)

- È garantito che esista un corridoio tra le stanze  $m$  e  $t$ .



### Subtask 3 (20 punti)

- $n \leq 1000$

### Subtask 4 (35 punti)

- nessuna limitazione aggiuntiva

## Output

Stampa il numero di mosse di Dumbo.

## Esempio

### Input

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

### Output

4

## Nota

Un possibile scenario:

- Dumbo blocca il passaggio tra le stanze 4 e 7.
- Il topolino si sposta nella stanza 6. Il passaggio tra le stanze 4 e 6 è sporco.
- Dumbo blocca il passaggio tra le stanze 6 e 8.
- Il topolino non può muoversi.
- Dumbo pulisce il passaggio tra le stanze 4 e 6.
- Il topolino si sposta nella stanza 4. Il passaggio tra le stanze 4 e 6 è sporco.
- Dumbo blocca il passaggio tra le stanze 2 e 3.
- Il topolino si sposta nella stanza 2. Il passaggio tra le stanze 2 e 4 è sporco.
- Dumbo non fa nulla.
- Il topolino non può che muoversi nella stanza 1 e rimane intrappolato.

Dumbo ha fatto 4 mosse.



## Sure Bet

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

La fortuna è fondamentale per far scommesse. Alcune persone migliorano le proprie possibilità e i propri guadagni avendo una buona conoscenza del soggetto della scommessa. Noi adotteremo un approccio differente.

Molti siti di scommesse offrono diverse *quote* per lo stesso evento (una *quota* di  $x$  significa che se scommetti 1€ ed hai indovinato il risultato correttamente, ottieni  $x$ € indietro. Se invece non indovini il risultato non ottieni niente indietro. Nota che paghi 1€ indipendentemente dal risultato). E se potessi essere sicuro di ottenere un profitto sicuro piazzando correttamente diverse scommesse? Il tuo obiettivo sarebbe quello di rendere questo profitto più grande possibile. Sei in questa particolare situazione.

L'evento sul quale vogliamo farti scommettere può avere due diversi esiti. Ci sono  $n$  siti di scommesse che offrono quote differenti: denotiamo la quota offerta dall' $i$ -esimo sito di scommesse sul primo esito con  $a_i$  e sul secondo esito con  $b_i$ . Puoi piazzare una scommessa su un qualsiasi sottoinsieme delle quote offerte, puoi anche scommettere su entrambi i risultati presso lo stesso sito di scommesse. Comunque, tutte le scommesse devono essere di esattamente 1€ e non puoi effettuare scommesse multiple sullo stesso risultato presso lo stesso sito.

Nel caso in cui avvenga il primo risultato, riceverai  $a_i$ € da ogni sito di scommesse  $i$  presso il quale hai piazzato una scommessa sul primo risultato. Analogamente, nel caso del secondo risultato, riceverai  $b_i$ € da tutti i siti di scommesse presso i quali hai piazzato una scommessa sul secondo risultato. Ovviamente, in entrambi i casi, hai già pagato 1€ per ogni scommessa che hai fatto.

Qual è il più grande profitto *garantito* (ovvero che ottieni indipendentemente dal risultato) se piazzhi tutte le tue scommesse in modo ottimale?

### Limiti

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

### Subtask 1 (20 punti)

- $n \leq 10$

### Subtask 2 (40 punti)

- $n \leq 1\,000$

### Subtask 3 (40 punti)

- nessuna limitazione aggiuntiva

### Input

La prima riga contiene il numero di siti di scommesse  $n$ . Le  $n$  righe successive descrivono le quote offerte da ogni sito di scommesse: due numeri reali  $a_i$  e  $b_i$  separati da uno spazio, che rappresentano le quote per il primo e il secondo risultato offerte dall' $i$ -esimo sito di scommesse. Le quote saranno date con al massimo 4 cifre decimali.



## Output

Stampa il massimo profitto garantito arrotondato ad un numero con esattamente 4 cifre decimali. Di seguito i comandi per stampare i numeri in virgola mobile (*floating point*) nei diversi linguaggi supportati:

- C e C++: `printf("%.4lf",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));`

## Esempio

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

## Nota

La strategia ottimale per la scommessa consiste nello scommettere sul secondo risultato presso il primo sito di scommesse e sul primo risultato presso il terzo e quarto sito di scommesse. Nel caso del primo risultato, avrai guadagnato  $1.6 + 1.9 - 3\text{€} = 0.5\text{€}$  e, nel caso del secondo risultato, avrai guadagnato  $3.7 - 3\text{€} = 0.7\text{€}$ . Quindi hai 50 centesimi garantiti, qualunque sia il risultato.