

Problema 3 - Shuffle

100 puncte

Fie G un graf orientat aciclic fără costuri pe muchii. În această problemă vom analiza ce se întâmplă dacă folosim o parcurgere în adâncime pentru a calcula drumul de lungime minimă dintre nodul 1 și nodul N . Mai exact, vom rula algoritmul descris de următoarea secvență de pseudocod:

```
viz[x] = 0, oricare ar fi x  
dist[1] = 0
```

```
DFS(nod):  
    viz[nod] = 1  
    pentru toti vecinii v din lista de adiacență a lui nod:  
        daca viz[v] este 0:  
            dist[v] = dist[nod] + 1  
            DFS(v)
```

```
DFS(1)  
afișează dist[N]
```

Formarea listelor de adiacență ale grafului urmează următorul pseudocod (unde $lista[x]$ reprezintă lista vecinilor lui x) :

```
lista[x] = [], oricare ar fi x
```

```
citeste n, m
```

```
pentru i de la 1 la m:  
    citeste a, b  
    adauga b la sfarsitul lui lista[a]
```

Bineînțeles, acest algoritm nu este întotdeauna corect, deoarece distanța calculată depinde de ordinea în care sunt procesați vecinii unui nod în timpul citirii.

De exemplu, dacă la construirea grafului se citește întâi muchia $(1 \rightarrow 2)$ și apoi muchia $(1 \rightarrow 3)$, atunci când vecinii lui 1 sunt prelucrați, primul vecin va fi 2, iar următorul va fi 3. Așadar, distanța calculată cu primul algoritm de mai sus poate fi diferită, în funcție de ordinea în care sunt adăugate muchiile de la citire.

Prin urmare, pentru un graf dat, suntem curioși pentru câte dintre cele $M!$ permutări ale muchilor lui G , algoritmul DFS va da distanța minimă.

Date de intrare

Pe prima linie a fișierului de intrare `shuffle.in` se vor afla două numere naturale N , numărul de noduri, și M , numărul de muchii. Pe următoarele M linii se vor afla două numere x și y , reprezentând faptul că în graf există muchia orientată de la x la y .

Date de ieșire

Pe prima linie a fișierului de ieșire `shuffle.out` se va afișa un singur număr, care reprezintă numărul de permutări acceptabile modulo **1 000 000 007**.

Restricții și precizări

- $1 \leq N \leq 100\ 000$
- $1 \leq M \leq 200\ 000$
- Se garantează că pentru teste în valoare de 10% din punctaj, $N, M \leq 10$
- Se garantează că pentru teste în valoare de 25% din punctaj, $N \leq 20, M \leq 60$
- Se garantează că pentru teste în valoare de 60% din punctaj, $N \leq 1400, M \leq 4000$
- Se garantează că o muchie nu va apărea de două ori în fișierul de intrare.
- Se garantează că fiecare muchie aparține cel puțin unui drum de la **1** la **N**.

Exemple

shuffle.in	shuffle.out	Explicații
3 3 1 2 1 3 2 3	3	Cele trei permutări pentru care se obține distanța minimă sunt: (1→3) (1→2) (2→3) (1→3) (2→3) (1→2) (2→3) (1→3) (1→2)
4 5 1 2 1 3 2 3 2 4 3 4	90	Distanța minimă de la 1 la 4 este 2 și sunt 90 de permutări ale celor 5 muchii care dau această distanță minimă.
7 11 2 4 3 2 1 3 2 6 6 7 4 5 4 6 6 5 5 7 2 5 3 4	24948000	

Timp maxim de execuție/test: 1.5 secunde

Memorie totală disponibilă: 64 MB, din care 64 MB pentru stivă

Dimensiune maximă a sursei: 20 KB