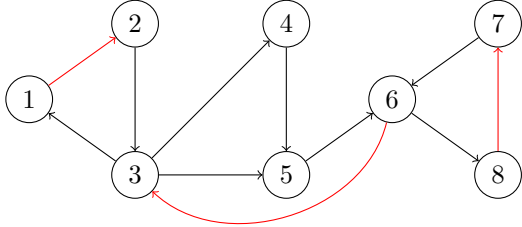


Acyklizacja (acyklizacja)

Memory limit: 64 MB

Time limit: 1.00 s

Jasio uwielbia grafy skierowane (w których krawędzie bieżą tylko w jedną stronę). A acykliczne (bez cykli) to już w ogóle najbardziej. Niestety, na urodziny otrzymał graf skierowany, ale niekoniecznie acykliczny. Darowanemu koniowi nie zagląda się w zęby, ale może da się zmazać niezbyt wiele krawędzi, żeby graf był acykliczny?



Napisz program, który: wczyta opis grafu, wyznaczy mały zbiór krawędzi, którego usunięcie z grafu jest wystarczające do osiągnięcia acykliczności i wypisze wynik na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby naturalne N oraz M , oddzielone pojedynczym odstępem i określające liczbę wierzchołków i krawędzi grafu. W kolejnych M wierszach znajduje się opis krawędzi grafu, po jednej w wierszu. Opis każdej krawędzi składa się z dwóch liczb naturalnych u_i oraz v_i określających istnienie krawędzi w jedną stronę z u_i do v_i .

W grafie na wejściu nie ma pętli ani krawędzi wielokrotnych.

Wyjście

W pierwszym wierszu wyjścia powinna się znaleźć jedna liczba naturalna $R \leq \frac{M}{2}$ określająca liczbę krawędzi do usunięcia. W kolejnych R wierszach należy wypisać usuwane krawędzie (w formacie jak na wejściu). Każda usuwana krawędź ma pojawić się na wyjściu dokładnie raz.

Pozostałe w grafie krawędzie (te, które pojawiły się na wejściu, ale nie pojawiły na wyjściu) powinny tworzyć graf acykliczny.

Jeśli istnieje wiele możliwych rozwiązań, należy wypisać dowolne z nich. Zauważ, że nie jest wymagane usunięcie jak najmniejszej liczby krawędzi, a jedynie zmieszczenie się w limicie $\frac{M}{2}$.

Ograniczenia

$1 \leq N \leq 100\,000$, $0 \leq M \leq 200\,000$.

Przykład

Input	Output	Explanation
8 11	4	Obrazek w treści przedstawia sytuację z testu przykładowego.
1 2	3 1	
2 3	6 3	
3 1	8 7	
3 4	7 6	
4 5		
3 5		
5 6		
6 3		
6 8		
8 7		
7 6		