

Задача А. Слово и ДКА

Имя входного файла: `problem1.in`
Имя выходного файла: `problem1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан детерминированный конечный автомат и слово. Определить, допускает ли данный ДКА заданное слово.

Формат входного файла

В первой строке входного файла находится слово, состоящее из не более чем 100000 строчных латинских букв.

Во второй строке содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно. ($1 \leq n, m \leq 100000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Требуется выдать строку “Accepts”, если автомат принимает заданное слово, или “Rejects” в противном случае.

Пример

problem1.in	problem1.out
abacaba 2 3 1 2 1 2 a 2 1 b 2 1 c	Accepts

Задача В. Слово и НКА

Имя входного файла: `problem2.in`
Имя выходного файла: `problem2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан недетерминированный конечный автомат и слово. Определить, допускает ли данный НКА заданное слово.

Формат входного файла

В первой строке входного файла находится слово, состоящее из не более чем 10000 строчных латинских букв.

Во второй строке содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно ($1 \leq n \leq 100$, $1 \leq m \leq 1000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1.

Формат выходного файла

Требуется выдать строку “Accepts”, если автомат принимает заданное слово, или “Rejects” в противном случае.

Пример

<code>problem2.in</code>	<code>problem2.out</code>
abacaba	Accepts
4 6 1	
2	
1 2 a	
2 1 c	
2 3 b	
3 2 a	
2 4 b	
1 4 a	

Задача С. Количество слов в языке

Имя входного файла: `problem3.in`
Имя выходного файла: `problem3.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан детерминированный конечный автомат. Требуется определить количество допускаемых им слов по модулю $10^9 + 7$

Формат входного файла

В первой строке содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно ($1 \leq n, m \leq 100000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Выведите количество слов, допускаемых автоматом по модулю $10^9 + 7$. Если таких слов существует бесконечно много, требуется вывести -1

Примеры

<code>problem3.in</code>	<code>problem3.out</code>
<code>1 1 1</code> <code>1</code> <code>1 1 a</code>	<code>-1</code>
<code>3 5 1</code> <code>3</code> <code>1 2 a</code> <code>1 2 b</code> <code>2 3 a</code> <code>2 3 b</code> <code>2 3 c</code>	<code>6</code>

Задача D. Число слов длины l в языке ДКА

Имя входного файла: `problem4.in`
Имя выходного файла: `problem4.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан детерминированный конечный автомат и число l . Требуется определить количество допускаемых им слов длины l по модулю $10^9 + 7$

Формат входного файла

В первой строке содержатся числа n , m , k и l — количество состояний, переходов и допускающих состояний в автомате, а также длина слов ($1 \leq n, m \leq 100$, $1 \leq k \leq n$, $1 \leq l \leq 10^3$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Выведите количество слов длины l , допускаемых автоматом, по модулю $10^9 + 7$.

Пример

<code>problem4.in</code>	<code>problem4.out</code>
3 6 1 1	1
3	
1 2 a	
1 2 b	
2 3 a	
2 3 b	
2 3 c	
1 3 q	

Задача Е. Число слов длины l в языке НКА

Имя входного файла: `problem5.in`
Имя выходного файла: `problem5.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задан недетерминированный конечный автомат и число l . Требуется определить количество допустимых им слов длины l по модулю $10^9 + 7$

Формат входного файла

В первой строке содержатся числа n , m , k и l — количество состояний, переходов и допускающих состояний в автомате, а также длина слов ($1 \leq n, m \leq 100$, $1 \leq k \leq n$, $1 \leq l \leq 10^3$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате " $a b c$ ", где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что алгоритм Томпсона найдёт такой детерминированный автомат, распознающий тот же язык, имеющий не более 100 состояний.

Формат выходного файла

Требуется выдать количество слов длины l , допускаемых автоматом, по модулю $10^9 + 7$.

Пример

<code>problem5.in</code>	<code>problem5.out</code>
3 6 1 1	1
3	
1 2 a	
1 2 b	
2 3 a	
2 3 b	
2 3 c	
1 3 q	

Задача F. Изоморфизм ДКА

Имя входного файла: `isomorphism.in`
Имя выходного файла: `isomorphism.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Задано два детерминированных конечных автомата. Определить, изоморфны ли они друг другу. Гарантируется, что все состояния автоматов достижимы.

Формат входного файла

Во входном файле находятся два описания ДКА. Формат описания следующий:

Во первой строке описания содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно. ($1 \leq n, m \leq 100000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате " $a b c$ ", где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Требуется выдать строку "YES", если автоматы изоморфны, или "NO" в противном случае.

Пример

<code>isomorphism.in</code>	<code>isomorphism.out</code>
3 3 1 3 1 2 a 1 3 c 2 3 b 3 3 1 2 1 3 a 1 2 c 3 2 b	YES

Примечание

Автоматы называются изоморфными, если существует биекция между их вершинами такая, что сохраняются все переходы, терминальные состояния соответствуют терминальным а начальные — начальным

Задача G. Эквивалентность ДКА

Имя входного файла: `equivalence.in`
Имя выходного файла: `equivalence.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Задано два детерминированных конечных автомата. Определить, эквивалентны ли они друг другу.

Формат входного файла

Во входном файле находятся два описания ДКА. Формат описания следующий:

Во первой строке описания содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно. ($1 \leq n, m \leq 1000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Требуется выдать строку “YES”, если автоматы эквивалентны, или “NO” в противном случае.

Пример

<code>equivalence.in</code>	<code>equivalence.out</code>
1 1 1 1 1 1 a 2 2 2 1 2 1 2 a 2 2 a	YES

Примечание

Автоматы называются эквивалентными, если они допускают один и тот же язык

Задача Н. Минимизация ДКА

Имя входного файла: `minimization.in`
Имя выходного файла: `minimization.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Задан детерминированный конечный автомат. Требуется построить эквивалентный ему автомат с минимальным количеством состояний.

Формат входного файла

Во первой строке входного файла содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно. ($1 \leq n, m \leq 1000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Требуется выдать результирующий автомат в том же формате.

Пример

<code>minimization.in</code>	<code>minimization.out</code>
2 2 2	1 1 1
1 2	1
1 2 a	1 1 a
2 2 a	

Примечание

В следующей задаче требуется сделать то же самое, но с более жесткими ограничениями.

Задача I. Быстрая минимизация ДКА

Имя входного файла: `fastminimization.in`
Имя выходного файла: `fastminimization.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайта

Задан детерминированный конечный автомат. Требуется построить эквивалентный ему автомат с минимальным количеством состояний.

Формат входного файла

Во первой строке входного файла содержатся числа n , m и k — количество состояний, переходов и допускающих состояний в автомате соответственно. ($1 \leq n, m \leq 50000$, $1 \leq k \leq n$).

В следующей строке содержатся k чисел — номера допускающих состояний (состояния пронумерованы от 1 до n).

В следующих m строках описываются переходы в формате “ $a b c$ ”, где a — номер исходного состояния перехода, b — номер состояния, в которое осуществляется переход и c — символ (строчная латинская буква), по которому осуществляется переход.

Стартовое состояние автомата всегда имеет номер 1. Гарантируется, что из любого состояния не более одного перехода по каждому символу.

Формат выходного файла

Требуется выдать результирующий автомат в том же формате.

Пример

<code>fastminimization.in</code>	<code>fastminimization.out</code>
<code>2 2 2</code>	<code>1 1 1</code>
<code>1 2</code>	<code>1</code>
<code>1 2 a</code>	<code>1 1 a</code>
<code>2 2 a</code>	