

Задача 1. Изоморфизм ДКА

Имя входного файла: `isomorphism.in`
 Имя выходного файла: `isomorphism.out`
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 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

isomorphism.in	isomorphism.out
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 2. Эквивалентность ДКА

Имя входного файла: `equivalence.in`
Имя выходного файла: `equivalence.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 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

equivalence.in	equivalence.out
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 3. Минимизация ДКА

Имя входного файла: `minimization.in`
Имя выходного файла: `minimization.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 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 4. Быстрая минимизация ДКА

Имя входного файла: `fastminimization.in`
Имя выходного файла: `fastminimization.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 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>
2 2 2	1 1 1
1 2	1
1 2 a	1 1 a
2 2 a	