

## Задача А. Изоморфизм ДКА

Имя входного файла: isomorphism.in  
Имя выходного файла: isomorphism.out  
Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

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

### Формат входных данных

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. ( $1 \leq n, m \leq 10^5$ ,  $1 \leq k \leq n$ ).

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 выходных данных

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

### Примеры

| isomorphism.in   | isomorphism.out |
|--|-----------------|
| 3 3 1<br>3<br>1 2 a<br>1 3 c<br>2 3 b<br>3 3 1<br>2<br>1 3 a<br>1 2 c<br>3 2 b | YES             |

### Замечание

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

## Задача В. Эквивалентность ДКА

Имя входного файла: `equivalence.in`  
Имя выходного файла: `equivalence.out`  
Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

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

### Формат входных данных

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 выходных данных

Требуется выдать строку «YES», если автоматы эквивалентны, или «NO» в противном случае.

### Примеры

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

### Замечание

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

## Задача С. Минимизация ДКА

Имя входного файла:            `minimization.in`  
Имя выходного файла:           `minimization.out`  
Ограничение по времени:       2.5 секунд  
Ограничение по памяти:         256 мегабайт

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

### Формат входных данных

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 выходных данных

Требуется выдать результирующий автомат в том же формате.

### Примеры

| <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                        |                               |

### Замечание

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

## Задача D. Быстрая минимизация ДКА

Имя входного файла: `fastminimization.in`  
Имя выходного файла: `fastminimization.out`  
Ограничение по времени: 2.5 секунд  
Ограничение по памяти: 256 мегабайт

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

### Формат входных данных

Во первой строке входного файла содержатся числа  $n$ ,  $m$  и  $k$  — количество состояний, переходов и допускающих состояний в автомате соответственно ( $1 \leq n, m \leq 50\,000$ ,  $1 \leq k \leq n$ ).

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 выходных данных

Требуется выдать результирующий автомат в том же формате.

### Примеры

| <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                            |                                   |

## Задача Е. Посвят

|                         |                   |
|-------------------------|-------------------|
| Имя входного файла:     | стандартный ввод  |
| Имя выходного файла:    | стандартный вывод |
| Ограничение по времени: | 2.5 секунд        |
| Ограничение по памяти:  | 256 мегабайт      |

Для проведения посвята в июльской смене ЛКШ 2017 преподавы решили построить лабиринт. Он будет состоять из  $n$  круглых комнат, которые соединены узкими коридорами. Однако посвят уже завтра, и преподавы не успеют построить целых  $n$  комнат!

Однако они заметили, что некоторые комнаты могут быть совершенно идентичными. Это позволяет уменьшить размер лабиринта так, что никто не заметит.

Комнаты  $A$  и  $B$  называются совершенно идентичными, если школьник, которого завели в одну из них, не сможет определить, в какую именно, лишь исследуя лабиринт. При этом все коридоры неотличимы, а двери во всех комнатах расположены через равные промежутки, поэтому комнаты можно отличить только по количеству дверей. Естественно, при входе в комнату школьник знает, через какой коридор он только что вошёл. Когда школьник только попадает в лабиринт, он стоит в самом центре комнаты и смотрит в произвольном направлении.

Требуется вывести все максимальные по включению множества совершенно идентичных комнат.

### Формат входных данных

В первой строке задано число  $n$  — количество комнат ( $1 \leq n \leq 100$ ).

В  $i$ -й из следующих  $n$  строк описана  $i$ -я комната. Сначала идёт число  $k$  — количество дверей, а затем  $k$  различных чисел — номера комнат, куда ведут двери в порядке обхода по часовой стрелке с произвольного места ( $0 \leq k < n$ ).

Комнаты нумеруются с единицы. Если из  $a$  есть дверь в  $b$ , то из  $b$  есть дверь в  $a$ . Нет коридоров из комнаты в неё же. Конечно же, выходов из лабиринта тоже нет.

### Формат выходных данных

Для каждого максимального по включению множества совершенно идентичных комнат, кроме множеств размера 1, выведите в отдельной строке номера входящих в него комнат в возрастающем порядке. Множества следует упорядочить по первому номеру.

Если таких множеств нет, выведите «none».

## Примеры

| стандартный ввод  | стандартный вывод               |
|---|---------------------------------|
| 13<br>2 2 4<br>3 1 3 5<br>2 2 4<br>3 1 3 6<br>2 2 6<br>2 4 5<br>2 8 9<br>2 7 9<br>2 7 8<br>2 11 13<br>2 10 12<br>2 11 13<br>2 10 12 | 2 4<br>5 6<br>7 8 9 10 11 12 13 |
| 6<br>3 3 4 5<br>0<br>1 1<br>1 1<br>2 1 6<br>1 5   | none                            |