

## Задача А. Минимальный каркас

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

Требуется найти в связном графе остовное дерево минимально веса.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $1 \leq n \leq 20000$ ,  $0 \leq m \leq 100000$ ). Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100000$ ).

Граф является связным.

### Формат выходного файла

Выведите единственное целое число - вес минимального остовного дерева.

### Примеры

<code>mst.in</code>	<code>mst.out</code>
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

## Задача В. Мосты

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

Дан неориентированный граф. Требуется найти все мосты в нём.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество мостов в заданном графе. На следующей строке выведите  $b$  целых чисел — номера рёбер, которые являются мостами, в возрастающем порядке. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

### Примеры

<code>bridges.in</code>	<code>bridges.out</code>
6 7	1
1 2	3
2 3	
3 4	
1 3	
4 5	
4 6	
5 6	

## Задача С. Точки сочленения

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

Дан неориентированный граф. Требуется найти все точки сочленения в нём.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

### Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество точек сочленения в заданном графе. На следующей строке выведите  $b$  целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

### Примеры

<code>points.in</code>	<code>points.out</code>
6 7	2
1 2	2
2 3	3
2 4	
2 5	
4 5	
1 3	
3 6	

## Задача D. День Объединения

Имя входного файла:           unionday.in  
Имя выходного файла:         unionday.out  
Ограничение по времени:       2 секунды  
Ограничение по памяти:         64 мегабайта

В Байтландии есть целых  $n$  городов, но нет ни одной дороги. Король страны, Вальдемар де Беар, решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было добраться от любого города до любого другого. Когда строительство будет завершено, король планирует отпраздновать День Объединения. К сожалению, казна Байтландии почти пуста, поэтому король требует сэкономить деньги, минимизировав суммарную длину всех построенных дорог.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 5\,000$ ) — количество городов в Байтландии. Каждая из следующих  $n$  строк содержит по два целых числа  $x_i, y_i$  — координаты  $i$ -го города ( $-10\,000 \leq x_i, y_i \leq 10\,000$ ). Никакие два города не расположены в одной точке.

### Формат выходного файла

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с точностью не менее  $10^{-3}$ .

### Пример

unionday.in	unionday.out
6 1 1 7 1 2 2 6 2 1 3 7 3	9.6568542495

## Задача Е. Противопожарная безопасность

Имя входного файла: `firesafe.in`  
Имя выходного файла: `firesafe.out`  
Ограничение по времени: 0.5 секунда  
Ограничение по памяти: 64 мегабайта

В Судиславле  $n$  домов. Некоторые из них соединены дорогами с односторонним движением.

В последнее время в Судиславле участились случаи пожаров. В связи с этим жители решили построить в посёлке несколько пожарных станций. Но возникла проблема: едущая по вызову пожарная машина, конечно, может игнорировать направление движения текущей дороги, однако возвращающаяся с задания машина обязана следовать правилам дорожного движения (жители Судиславля свято чтут эти правила!).

Ясно, что, где бы ни оказалась пожарная машина, у неё должна быть возможность вернуться на ту пожарную станцию, с которой она выехала. Но строительство станций стоит больших денег, поэтому на совете посёлка было решено построить минимальное количество станций таким образом, чтобы это условие выполнялось. Кроме того, для экономии было решено строить станции в виде пристроек к уже существующим домам.

Ваша задача — написать программу, рассчитывающую оптимальное положение станций.

### Формат входного файла

В первой строке входного файла задано число  $n$  ( $1 \leq n \leq 3000$ ). Во второй строке записано количество дорог  $m$  ( $1 \leq m \leq 100\,000$ ). Далее следует описание дорог в формате  $a_i b_i$ , означающее, что по  $i$ -й дороге разрешается движение автотранспорта от дома  $a_i$  к дому  $b_i$  ( $1 \leq a_i, b_i \leq n$ ).

### Формат выходного файла

В первой строке выведите минимальное количество пожарных станций  $K$ , которое необходимо построить. Во второй строке выведите  $K$  чисел в произвольном порядке — дома, к которым необходимо пристроить станции. Если оптимальных решений несколько, выведите любое.

### Примеры

<code>firesafe.in</code>	<code>firesafe.out</code>
5	2
7	4 5
1 2	
2 3	
3 1	
2 1	
2 3	
3 4	
2 5	

## Задача F. АлгоЛэнд

Имя входного файла: `algoland.in`  
Имя выходного файла: `algoland.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 32 мегабайта

Недавно королева страны AlgoLand придумала новый способ отмывания денег для своего королевского двора. Она решила, что всякий житель, желающий совершить путешествие из одного города страны в другой, должен расплатиться за это желание своими деньгами.

В стране AlgoLand есть  $N$  городов, пронумерованных от 1 до  $N$ . Некоторые города соединены дорогами, движение по которым разрешено в двух направлениях. Начиная движение по какой-нибудь дороге, путешественник обязательно должен доехать до ее конца.

Предположим теперь, что житель страны хочет совершить путешествие из города  $A$  в город  $B$ . Новый указ королевы гласит, что при проезде по любой дороге страны во время этого путешествия, полицейские могут взять с этого жителя таможенную пошлину в пользу королевского двора (а могут и не взять). Если при этом у жителя недостаточно денег для уплаты пошлины, то он автоматически попадает в тюрьму. Указ также устанавливает величину пошлины для каждой дороги страны. Так как королева заботится о жителях своей страны, то она запретила полицейским брать с жителя пошлину более чем два раза во время одного путешествия.

Отметим, что если существует несколько способов попасть из города  $A$  в город  $B$ , то житель может выбрать для путешествия любой из них по собственному желанию.

Напишите программу, которая определяет, какую минимальную сумму денег должен взять с собой житель, чтобы гарантированно не попасть в тюрьму во время путешествия.

### Формат входного файла

Первая строка входного файла содержит числа  $N$  и  $M$  ( $2 \leq N \leq 10^4$ ,  $1 \leq M \leq 10^5$ ), разделенные пробелом — количества городов и дорог. Следующие  $M$  строк описывают дороги. Каждая из этих строк описывает одну дорогу и содержит три числа  $X, Y, Z$  ( $1 \leq X, Y \leq N$ ,  $X \neq Y$ ,  $1 \leq Z \leq 10^9$ ) разделенных пробелами, означающие, что дорога соединяет города  $X$  и  $Y$  и пошлина за ее проезд равна  $Z$  денежных единиц. Последняя строка содержит числа  $A$  и  $B$  ( $1 \leq A, B \leq N$ ,  $A \neq B$ ) - номера начального и конечного городов путешествия. Гарантируется, что существует хотя бы один способ проезда из  $A$  в  $B$ .

### Формат выходного файла

Единственная строка выходного файла должна содержать одно число, равное минимальной сумме денег, которую должен взять с собой житель, чтобы иметь возможность совершить путешествие из города  $A$  в город  $B$  и при этом гарантированно не попасть в тюрьму независимо от действий полицейских.

### Примеры

<code>algoland.in</code>	<code>algoland.out</code>
5 6 1 5 1 5 4 1 5 2 2 4 2 1 3 2 1000000000 3 1 1000000000 1 3	1000000000