Firesafe. Противопожарная безопасность

Имя входного файла: `firesafe.in`
Имя выходного файла: `firesafe.out`

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 городе 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 выехала. Но строительство станций стоит больших денег, поэтому на совете города 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 ($1 \leqslant n \leqslant 3\,000$). Во второй строке записано количество дорог m ($1 \leqslant m \leqslant 100\,000$). 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 ($1 \leqslant a_i, b_i \leqslant n$).

Формат выходного файла

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

Пример

firesafe.in	firesafe.out
5 7 1 2 2 3 3 1 2 1 2 3 3 4 2 5	2 4 5

Domino. Марсианское домино

Имя входного файла: `domino.in`
Имя выходного файла: `domino.out`

Вернувшись домой с очередной экспедиции на Марс, Вася привез с собой найденный там чемодан с непонятными каменными прямоугольничками. Каждый из них имел длину в два раза большую, чем ширина, и был поделен чертой на 2 квадрата. В каждом квадрате нарисован какой-то значок. И ещё 1 значок нарисован на обратной стороне.

Вася долго бился над разгадкой тайны прямоугольничков. Однажды он проснулся посреди ночи от того, что наконец решил эту задачку. Оказывается, значки сверху — это цифры от 1 до n, а прямоугольнички — это фишки домино. Значок снизу — цена доминошки.

Теперь Васю мучает вопрос: можно ли составить цепочку по правилам игры в домино из всех этих фишек? К тому же он хочет, чтобы суммарная цена цепочки была минимальна. Но он очень хочет спать, поэтому сам не справится. Помогите Васе решить эту задачу.

Формат входного файла

Первая строка входного файла содержит натуральное число n — максимальное число, написанное на верхней поверхности фишки ($1 \leqslant n \leqslant 1\,000$). Далее следуют n строк, описывающих доминошки. В i-й из этих строк находится число m_i — количество доминошек, содержащих число i. Далее следуют m_i пар положительных чисел: первое — число, написанное на второй половине доминошки, а второе число — её цена.

Может быть несколько одинаковых доминошек, но нет ни одной, у которой с обоих концов написано одно и то же число.

Все числа во входном файле не превосходят 10^5; гарантируется, что общее число доминошек также не превосходит 10^5.

Формат выходного файла

Если решение существует, то в первую строку выходного файла выведите одно число — количество доминошек в искомой цепочке (считая первую и последнюю), а во вторую — числа, в порядке, в котором они будут лежать в цепи (место соединения доминошек выводить как одно число — см. пример).

Если решений нет, выведите в выходной файл одно число -1.

Если решений несколько, выведите любое.

Пример

domino.in	domino.out
4 2 2 1 2 2 4 1 2 4 4 3 5 1 1 2 2 5 4 8 2 3 8 2 4	5 1 2 3 4 2 1

Points. Точки сочленения

Имя входного файла: `points.in`
Имя выходного файла: `points.out`

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

Формат входного файла

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

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

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.

Пример

points.in	points.out
9 12 1 2 2 3 4 5 2 6 2 7 8 9 1 3 1 4 1 5 6 7 3 8 3 9	3 1 2 3

Bridges. Мосты

Имя входного файла: `bridges.in`
Имя выходного файла: `bridges.out`

Дан неориентированный граф. Требуется найти все мосты в нем.

Формат входного файла

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

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

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 ребер, которые являются мостами, в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Пример

bridges.in	bridges.out
6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6	1 3

Contest. Контест!

Имя входного файла: `contest.in`
Имя выходного файла: `contest.out`

В Берляндии проходит традиционное ежегодное соревнование по спортивному программированию «Берланд Open». Как вам наверняка известно, завтра состоятся N четвертьфиналов, в которых примут участие $2N$ программистов, по двое в каждом четвертьфинале. После соревнований пройдет процедура награждения, на которую будут приглашены все N победителей.

К сожалению, отношения между программистами в Берляндии весьма натянуты. В частности, есть M пар участников, которые настолько друг друга не любят, что процедура награждения будет сорвана, если на нее попадут два участника из одной пары.

Васе выпала тяжелая доля вести процедуру награждения и развлекать толпу программистов. Он очень не хочет этого делать и потому очень надеется, что процедура награждения будет сорвана. Сегодня он сумел раздобыть секретный план распределения программистов по N четвертьфиналам. Помогите ему определить, есть ли хоть один шанс, что награждение таки состоится успешно.

Формат входного файла

Первая строка входного файла содержит два целых неотрицательных числа N и M ($0 \leqslant N \leqslant 10^5$, $0 \leqslant M \leqslant 10^6$). Далее следуют N строк по два числа в каждой — номера участников i-го четвертьфинала. Далее следует M строк, содержащих по два числа v, u каждая ($1 \leqslant v, u \leqslant 2N$), означающих, что v-тый и u-тый программисты конфликтуют и одновременное их попадание на процедуру награждения обозначает срыв этой процедуры. Одна и та же пара может быть упомянута несколько раз.

Программисты нумеруются натуральными числами от 1 до $2N$.

Формат выходного файла

Если есть хоть один шанс, что процедура награждения пройдет успешно, то выведите в выходной файл N чисел — номера программистов, которые должны будут участвовать в этой процедуре (если есть несколько вариантов успешного проведения процедуры, выведите любой).

Если выбрать бесконфликтных победителей невозможно, то выведите одно число -1.

Пример

contest.in	contest.out
3 4 2 3 1 4 5 6 2 4 1 2 3 6 1 5	3 4 5