

Задача А. Паросочетание

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

Дан двудольный невзвешенный граф. Необходимо найти максимальное паросочетание.

Формат входных данных

В первой строке входного файла три целых числа n , m и k ($1 \leq n, m \leq 200$, $1 \leq k \leq n \times m$) — количество чисел в первой и второй долях, а также число ребер соответственно. Далее следуют k строк, в каждой из которых два числа a_i и b_i , что означает ребро между вершиной с номером a_i первой доли и вершиной с номером b_i второй доли. Вершины в обеих долях нумеруются с единицы.

Формат выходных данных

В выходной файл выведите одно число — максимальное число ребер в паросочетании.

Примеры

matching.in	matching.out
3 3 5 1 1 1 3 2 1 2 2 3 2	3

Задача В. Покрытие путями

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

Задан ориентированный ациклический граф. Требуется определить минимальное количество непересекающихся путей, покрывающих все вершины.

Формат входных данных

Первая строка входного файла содержит n и m — количество вершин и ребер графа соответственно ($2 \leq n \leq 1000$, $0 \leq m \leq 10^5$). В следующих m строках содержатся по два числа: номера вершин u и v , которые соединяет ребро (u, v) .

Формат выходных данных

В первой строке выходного файла выведите натуральное число k — минимальное количество путей, необходимых, чтобы покрыть все вершины.

Примеры

<code>paths.in</code>	<code>paths.out</code>
3 3 1 3 3 2 1 2	1

Задача С. Минимальное вершинное покрытие

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

Требуется построить в двудольном графе минимальное вершинное покрытие, если задано максимальное паросочетание.

Формат входных данных

В первой строке файла заданы два числа m и n ($1 \leq m, n \leq 4000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leq K_i \leq n$) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входном файле не превосходит 500 000. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе — m чисел $0 \leq L_i \leq n$ — соответствующая i -й вершине первой доли вершина второй доли, или 0, если i -я вершина первой доли не входит в паросочетание.

Формат выходных данных

Первая строка содержит размер минимального вершинного покрытия. Вторая строка содержит число вершин первой доли S , после которого записаны S чисел — номера вершин первой доли, входящих в вершинное покрытие, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Пример

<code>vertexcover.in</code>	<code>vertexcover.out</code>
3 2	2
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

Задача D. День рождения

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

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \leq k \leq 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \leq m \leq 150$ и $0 \leq n \leq 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i -й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Формат выходных данных

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых, разделённые одним пробелом. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Числа в каждой из этих двух строк разделяются ровно одним пробелом и выводятся в порядке возрастания. Если максимальных наборов несколько, то выведите любой из них.

Разделяйте вывод для разных наборов входных данных одной пустой строкой.

Пример

<code>birthday.in</code>	<code>birthday.out</code>
2	4
2 2	2 2
1 2 0	1 2
1 2 0	1 2
3 2	
1 2 0	4
2 0	2 2
1 2 0	1 3
	1 2

Задача Е. Шахматная доска

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

Вася любит играть в необычные шахматы. Как-то раз, когда его братишка Коля был еще очень маленький, Вася вернулся из школы, и увидел, что его любимую шахматную доску кто-то перекрасил. Вася не сильно разозлился на Колю, потому что очень любил своего младшего брата. Так как у них дома были только черная и белая краски, каждая клетка доски была покрашена в один из этих двух цветов.

Вася решил исправить ошибку брата, он решил покрасить доску так, чтобы она снова стала шахматной. Но Вася почему-то подумал, что хочет красить только диагонали, причем сразу целиком. Вася решил не тратить много времени, поэтому его интересует способ покраски, который содержит наименьшее количество действий. За одно действие Вася может покрасить полностью какую-либо диагональ, в любой из двух цветов: черный или белый. Диагонали бывают двух типов, в зависимости от направления прямой, на которой лежит диагональ. Диагональ, которая лежит на прямой, направленной влево и вниз, является диагональю первого типа, а диагональ, которая лежит на прямой, направленной вправо и вниз, — второго.

Вам предстоит помочь Васе. Задано испорченное Колей шахматное поле. Вам необходимо определить, за какое минимальное количество действий Вася сможет перекрасить доску так, чтобы она стала шахматной.

Формат входных данных

В первой строке входного файла записаны два целых числа: n и m ($1 \leq n, m \leq 100$) — количество строчек и количество столбцов шахматного поля, соответственно.

В следующих n строках записано поле. Каждая строка содержит по m символов, и описывает одну строку шахматного поля. Символ `W` соответствует белой клетке, `B` — черной.

Формат выходных данных

В первой строке выходного файла выведите число p — минимальное количество действий, которое потребуется Васе, чтобы его доска снова стала шахматной.

В следующих p строках выведите описания действий. Каждое действие описывается тремя параметрами: тип диагонали, координаты некоторой покрашенной клетки и цвет. Тип диагонали — это число 1 или 2 — тип диагонали. Координаты клетки — это два целых числа: строка и столбец одной из клеток, которую покрасили этим действием. Цвет — это символ `W` или `B`, белый и черный соответственно.

Примеры

chess.in	chess.out
3 3 WBB BWB BBW	1 1 1 3 W
3 3 WBW WWB WWW	1 2 2 1 B
1 3 WWW	1 1 1 2 B

Задача F. King. Король

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

И было у мужика
три дочери...

В Тридесятом царстве, Тридевятиом государстве жил-был король. И было у короля n дочерей. В Тридесятом царстве жили n прекрасных юношей, и король знал, какие юноши нравятся каждой дочери (поскольку дочери были молодыми и безшабашными, то им могли нравиться несколько юношей одновременно).

Однажды король приказал своему советнику подобрать для каждой дочери прекрасного юношу, за которого та сможет выйти замуж. Советник выполнил приказ и подобрал для каждой дочери для замужества прекрасного юношу, который ей нравился. Разумеется, каждый юноша может жениться только на одной из дочерей.

Посмотрев на список женихов, король сказал: «Мне нравится этот список, но я хочу знать для каждой дочери список всех юношей, за которых она может выйти замуж. Разумеется, при этом все остальные дочери также должны сохранить возможность выйти замуж за юношей, которые им нравятся».

Эта задача оказалась для советника слишком сложной. Помогите ему избежать казни, решив ее.

Формат входных данных

Первая строка входного файла содержит число n — количество дочерей ($1 \leq n \leq 2000$). Следующие n строк содержат списки прекрасных юношей, которые нравятся дочерям. В начале идет k_i — количество юношей, которые нравятся i -ой дочери. Затем идут k_i чисел — номера юношей. Сумма k_i не превышает 200 000.

Последняя строка входного файла содержит список, составленный советником — n различных чисел от 1 до n : для каждой дочери — номер прекрасного юноши, за которого она может выйти замуж. Гарантируется, что список корректен — то есть каждой девушке нравится выбранный для нее юноша.

Формат выходных данных

Выходной файл должен содержать n строк. Для каждой девушки выведите l_i — количество различных юношей, за которых она может выйти замуж. После этого выведите l_i чисел — номера этих юношей в произвольном порядке.

Пример

king.in	king.out
4	2 1 2
2 1 2	2 1 2
2 1 2	1 3
2 2 3	1 4
2 3 4	
1 2 3 4	

Задача G. Эйлеров путь

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

Дан неориентированный связный граф, не более трех вершин имеет нечетную степень. Требуется определить, существует ли в нем путь, проходящий по всем ребрам.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество вершин графа ($1 \leq n \leq 1000$). Далее следуют n строк, задающих ребра. В i -ой из этих строк находится число m_i — количество ребер, инцидентных вершине i . Далее следуют m_i натуральных чисел — номера вершин, в которые идет j -ое ребро из i -ой вершины.

Граф может содержать кратные ребра, но не содержит петель.

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число k — количество ребер в искомом маршруте, а во вторую $k + 1$ чисел — номера вершин в порядке их посещения.

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

Если решений несколько, выведите любое.

Примеры

euler.in	euler.out
4	5
2 2 2	1 2 3 4 2 1
4 1 4 3 1	
2 2 4	
2 3 2	

Задача Н. Разбиение на пути

Имя входного файла: `paths.in`
Имя выходного файла: `paths.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Разбейте рёбра неориентированного графа на минимальное количество рёберно-простых путей.

Формат входных данных

Дан граф. На первой строке число вершин n ($1 \leq n \leq 20\,000$) и число рёбер m ($1 \leq m \leq 20\,000$). Следующие m строк содержат описание рёбер графа. Каждая строка по два числа a_i и b_i ($1 \leq a_i, b_i \leq n$). Между каждыми двумя вершинами не более одного ребра. Граф связан.

Формат выходных данных

На первой строке минимальное число путей. На каждой следующей описании очередного пути – номера вершин в порядке прохождения.

Примеры

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

Задача I. Кодовый замок

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

Петя опоздал на тренировку по программированию! Поскольку тренировка проходит в воскресенье, главный вход в учебный корпус, где она проходит, оказался закрыт, а вахтёр ушёл куда-то по своим делам. К счастью, есть другой способ проникнуть в здание — открыть снаружи боковую дверь, на которой установлен кодовый замок.

На пульте замка есть d кнопок с цифрами $0, 1, \dots, d-1$. Известно, что код, открывающий замок, состоит из k цифр. Замок открывается, если последние k нажатий кнопок образуют код.

Поскольку Петя не имеет понятия, какой код открывает замок, ему придётся перебрать все возможные коды из k цифр. Но, чтобы как можно скорее попасть на тренировку, нужно минимизировать количество нажатий на кнопки. Помогите Пете придумать такую последовательность нажатий на кнопки, при которой все возможные коды были бы проверены, а количество нажатий при этом оказалось бы минимально возможным.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа d и k — количество кнопок на пульте и размер кода, соответственно ($2 \leq d \leq 10$, $1 \leq k \leq 20$).

Формат выходных данных

В первой строке выходного файла выведите искомую последовательность. Если последовательностей минимальной длины, перебирающих все возможные коды, несколько, можно выводить любую из них. Гарантируется, что d и k таковы, что минимальная длина последовательности не превосходит 1 мегабайта.

Примеры

<code>codelock.in</code>	<code>codelock.out</code>
2 3	0011101000
2 1	01