

Задача А. Компоненты реберной двусвязности

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

Компонентой реберной двусвязности графа $\langle V, E \rangle$ называется подмножество вершин $S \subset V$, такое что для любых различных u и v из этого множества существует не менее двух реберно не пересекающихся путей из u в v .

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

Формат входных данных

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

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

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент реберной двусвязности графа. Во второй строке выведите n натуральных чисел $a_1, a_2, \dots, a_n$, не превосходящих k , где a_i — номер компоненты реберной двусвязности, которой принадлежит i -я вершина.

Примеры

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

Задача В. Компоненты вершинной двусвязности

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

Компонентой вершинной двусвязности графа $\langle V, E \rangle$ называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

Формат входных данных

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

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

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент вершинной двусвязности графа. Во второй строке выведите m натуральных чисел $a_1, a_2, \dots, a_m$, не превосходящих k , где a_i — номер компоненты вершинной двусвязности, которой принадлежит i -е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Примеры

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

Задача С. Почтальон

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

В городе $\mathfrak{C}$ есть n площадей, соединенных улицами. При этом количество улиц не превышает ста тысяч и существует не более трех площадей, на которые выходит нечетное число улиц. Для каждой улицы известна ее длина. По улицам разрешено движение в обе стороны. В городе есть хотя бы одна улица. От любой площади до любой можно дойти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице так, чтобы длина его пути была наименьшей. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество площадей в городе ($1 \leq n \leq 1\,000$). Далее следуют n строк, задающих улицы. В i -й из этих строк находится число m_i — количество улиц, выходящих из площади i . Далее следуют m_i пар положительных чисел. В j -й паре первое число — номер площади, в которую идет j -я улица с i -й площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улиц с площади на нее саму.

Все числа во входном файле не превосходят 10^5 .

Формат выходных данных

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте (считая первую и последнюю), а во вторую — номера площадей в порядке их посещения.

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

Если решений несколько, выведите любое.

Примеры

<code>post.in</code>	<code>post.out</code>
4	5
2 2 1 2 2	1 2 4 3 2 1
4 1 2 4 4 3 5 1 1	
2 2 5 4 8	
2 3 8 2 4	

Задача D. 2-SAT

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

Формулировка 2-SAT: нужно подобрать значения n булевых переменных так, чтобы все m утверждений вида $x_{i_1} = e_1 \vee x_{i_2} = e_2$ обратились в истину. В данной задаче вам гарантируется, что решение существует.

Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных n и число утверждений m . Каждая из следующих m строк содержит числа i_1, e_1, i_2, e_2 , задает утверждение $x_{i_1} = e_1 \vee x_{i_2} = e_2$ ($0 \leq i_j < n$, $0 \leq e_j \leq 1$). Ограничения: сумма всех n не больше 100 000, сумма всех m не больше 300 000.

Формат выходных данных

Для каждого теста выведите строку из n нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

Примеры

2sat.in	2sat.out
1 0	0
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	

Задача Е. Олег Игоревич Мингалёв

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Олег Игоревич Мингалёв играет в игру «Кубб». В определенный момент игры ему нужно поднять лежащие на игровом поле чурбаны: каждый чурбан нужно поставить вертикально, и по правилам это можно сделать только поставив его в точку, где находился один из его концов.

Когда чурбаны будут подняты, противоположная команда будет их сбивать. И если чурбаны оказались стоящими близко, то их при должном умении легко сбить одним ударом. Поэтому Олегу Игоревичу важно максимизировать величину D : минимальное расстояние между какой-либо парой чурбанов (длина отрезка от основания первого до основания второго).

Помогите ему поставить каждый чурбан на один из своих концов так, чтобы величина D была максимальной.

Формат входных данных

В первой строке ввода дано число n ($2 \leq n \leq 100$) — число отрезков. Далее в каждой из n строк даны координаты концов отрезка в формате x_1, y_1, x_2, y_2 . Все координаты по модулю не превышают 10^9 . Все отрезки имеют ненулевую длину.

Формат выходных данных

В единственной строке выведите максимально возможную достижимую величину D — минимальное расстояние между выбранными концами отрезков, с точностью не менее 10^{-9} .

Примеры

стандартный ввод	стандартный вывод
2 0 0 2 0 3 0 4 0	4
2 0 0 1 1 1 2 2 2	2.8284271247461902909

Задача F. Таня и пароль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из $n + 2$ символов. Она выписала все возможные n трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось n бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 2 \cdot 10^5$), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие n строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

Формат выходных данных

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

Примеры

стандартный ввод	стандартный вывод
5 aca aba aba cab bac	YES abacaba
4 abc bCb cb1 b13	NO
7 aaa aaa aaa aaa aaa aaa aaa	YES aaaaaaaaa

Задача G. Студентам — бесплатно!

Имя входного файла:	<code>students-free.in</code>
Имя выходного файла:	<code>students-free.out</code>
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	64 мегабайта

Зал Большого галактического театра состоит из S рядов, по S мест в каждом ряду. Продажа билетов на каждый спектакль происходит по следующему принципу: первые $S^2 - N$ ценителей прекрасного приобретают билеты на любые места по их вкусу, а оставшиеся N кресел администрация бесплатно выделяет студентам, отдавая дань сложившимся традициям.

Во избежание обвинений в дискриминации по половому признаку, рассаживать студентов по этим N местам необходимо таким образом, чтобы:

- в каждом ряду количество девушек-студенток и количество юношей-студентов различалось бы не более чем на 1;
- на каждой "вертикали мест" (т. е. местах, имеющих один и тот же номер, но расположенных в разных рядах) количество девушек-студенток и количество юношей-студентов также различалось бы не более чем на 1.

Таким образом, после продажи билетов ценителям прекрасного билетёры должны распределить оставшиеся N кресел на женские и мужские с соблюдением этих правил. Каждое место в зале определяется двумя числами от 1 до S — номером ряда и номером самого места в этом ряду. Студенческое кресло номер i расположено в a_i -м ряду и имеет в нём номер b_i . Поскольку ценители прекрасного могли занять совершенно любые места, числа a_i и b_i могут принимать любые значения от 1 до S . В частности, может оказаться так, что в каком-нибудь ряду не будет ни одного студенческого места.

Ради упрощения работы билетёров администрация обращается к вам с заданием написать программу, которая автоматизирует процесс распределения студенческих мест на мужские и женские.

Формат входных данных

Сначала вводятся два целых числа S и N ($1 \leq S \leq 100\,000, 1 \leq N \leq \min\{100\,000, S^2\}$). Далее расположены N пар натуральных чисел (a_i, b_i) , не превосходящих S . Гарантируется, что все места различные.

Формат выходных данных

Если искомого способа не существует, выведите **Impossible**. Иначе выведите единственную строку из N символов M (мужское) и W (женское). Символ на i -й позиции соответствует статусу i -го места в той же нумерации, в которой они были перечислены во входных данных.

Примеры

<code>students-free.in</code>	<code>students-free.out</code>
2 2 2 1 1 2	WM
3 5 1 2 2 3 1 3 2 1 1 1	MMWWM