

Задача А. Поиск цикла

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

Формат входных данных

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 100\,000$, $M \leq 100\,000$) — количество вершин и рёбер в графе соответственно. Далее в M строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

Формат выходных данных

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

Примеры

stdin	stdout
2 2 1 2 2 1	YES 1 2
2 2 1 2 1 2	NO

Задача В. Конденсация графа

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Требуется найти количество ребер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных ребер.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leq 10\,000$, $m \leq 100\,000$). Следующие m строк содержат описание ребер, по одному на строке. Ребро номер i описывается двумя натуральными числами b_i , e_i — началом и концом ребра соответственно ($1 \leq b_i, e_i \leq n$). В графе могут присутствовать кратные ребра и петли.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — количество ребер в конденсации графа.

Пример

stdin	stdout
4 4 2 1 3 2 2 3 4 3	2

Задача С. Кратчайший путь коня

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На шахматной доске размером 8×8 заданы две клетки. Соедините эти клетки кратчайшим путем коня.

Формат входных данных

Программа получает на вход координаты двух клеток, каждая в отдельной строке. Координаты клеток задаются в виде буквы (от “a” до “h”) и цифры (от 1 до 8) без пробелов.

Формат выходных данных

Программа должна вывести путь коня, начинающийся и заканчивающийся в данных клетках и содержащий наименьшее число клеток.

Пример

stdin	stdout
a1	a1
h8	b3
	a5
	b7
	d8
	f7
	h8

Задача D. Получи дерево

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Дан связный неориентированный граф без петель и кратных ребер. Разрешается удалять из него ребра. Требуется получить дерево.

Формат входных данных

Сначала вводятся два числа: N (от 1 до 100) и M — количество вершин и ребер графа соответственно. Далее идет M пар чисел, задающих ребра. Гарантируется, что граф связный.

Формат выходных данных

Выведите $N - 1$ пару чисел — ребра, которые войдут в дерево. Ребра можно выводить в любом порядке.

Пример

stdin	stdout
4 4	1 2
1 2	2 3
2 3	3 4
3 4	
4 1	

Задача Е. Расстояние между вершинами

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Дан неориентированный взвешенный граф. Найти вес минимального пути между двумя вершинами.

Формат входных данных

Первая строка входного файла содержит натуральные числа N , M , вторая строка содержит натуральные числа S и F ($N \leq 5\,000$, $M \leq 100\,000$, $1 \leq S, F \leq N$, $S \neq F$) — количество вершин и ребер графа а также номера вершин, длину пути между которыми требуется найти.

Следующие M строк по три натуральных числа b_i , e_i и w_i — номера концов i -ого ребра и его вес соответственно ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$).

Формат выходных данных

Первая строка должна содержать одно натуральное число — вес минимального пути между вершинами S и F . Во второй строке через пробел выведите вершины на кратчайшем пути из S в F в порядке обхода.

Если путь из S в F не существует, выведите -1 .

Пример

stdin	stdout
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3 1 2 3

Задача F. Флойд

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами. Гарантируется, что в графе нет циклов отрицательного веса.

Формат входных данных

В первой строке вводится единственное число N ($1 \leq N \leq 100$) — количество вершин графа. В следующих N строках по N чисел задается матрица смежности графа (j -ое число в i -ой строке — вес ребра из вершины i в вершину j). Все числа по модулю не превышают 100. На главной диагонали матрицы — всегда нули.

Формат выходных данных

Выведите N строк по N чисел — матрицу расстояний между парами вершин, где j -ое число в i -ой строке равно весу кратчайшего пути из вершины i в j .

Пример

stdin	stdout
4	0 5 7 13
0 5 9 100	12 0 2 8
100 0 2 8	11 16 0 7
100 100 0 7	4 9 11 0
4 100 100 0	

Задача G. Лабиринт знаний

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Участникам сборов подарили билеты на аттракцион “Лабиринт знаний”. Лабиринт представляет собой N комнат, занумерованных от 1 до N , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате N . Каждый участник сборов проходит лабиринт ровно один раз и набирает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

Формат входных данных

Первая строка входного файла содержит целые числа N ($1 \leq N \leq 2\,000$) — количество комнат и M ($1 \leq M \leq 10\,000$) — количество дверей. В каждой из следующих M строк содержится описание двери — номера комнат, из которой она ведет и в которую она ведет (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10 000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери.

Формат выходных данных

В выходной файл выведите “:)” — если можно пройти лабиринт и получить неограниченно большой запас знаний, “:(” — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

Пример

stdin	stdout
2 2 1 2 5 1 2 -5	5

Дополнительные задачи

Задача Н. Hard Test

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Андрей Сергеевич занят подготовкой задач к командной олимпиаде ЛКШ. Решение одной из задач основано на алгоритме Дейкстры и А.С. хочет подготовить сложный тест для этого алгоритма.

Алгоритм Дейкстры действует следующим образом. Пусть G — это ориентированный граф с множеством вершин V , множеством ребер E и весовой функцией $w : E \rightarrow \mathbb{R}^+$. Пусть все вершины достижимы из вершины s . Алгоритм использует множество вершин U , изначально инициализированный как пустой. Каждая вершина помечена либо целым числом, либо $+\infty$. Изначально все вершины помечены $+\infty$, а вершина s помечена нулем. Обозначим метку вершины v как $d[v]$.

Шаг алгоритма следующий: выбирается вершина с минимальной меткой, не входящая в U . Пусть это вершина u . Вершина u добавляется в множество U и каждое ребро $uv \in E$ релаксируется. Релаксация заменяет $d[v]$ на $\min(d[v], d[u] + w(uv))$. Алгоритм заканчивается, когда все вершины добавлены в U . Если метка вершины v изменяется, релаксация называется активной.

Теперь Андрей хочет создать граф с N вершинами и M ребрами, в котором алгоритм Дейкстры сделает максимально возможное количество активных релаксаций. Помогите ему создать такой граф. Чтобы избежать неопределенности, пусть каждый раз существует ровно одна вершина, которая выбирается с минимальной меткой из вершин, которые не входят в U .

Формат входных данных

Первая строка входного файла содержит два целых числа: N и M — число вершин и ребер в графе, который Андрей хотел бы создать ($4 \leq N \leq 1000$, $N - 1 \leq M \leq N^2/5$).

Формат выходных данных

В выходной файл выведите M строк — рёбра графа. Каждая строка должна содержать три целых числа: начала ребра, конец ребра и его вес. Все веса должны быть неотрицательные и не должны превосходить 10^6 . Все вершины должны быть доступны из вершины с номером 1.

Если алгоритм Дейкстры начинает работу из вершины $s = 1$ он должен сделать максимальное возможное количество активных релаксаций среди всех графов с N вершинами и M ребрами. Граф не должен иметь петель и кратных рёбер.

Пример

stdin	stdout
4 3	1 2 0 1 3 1 1 4 2
5 5	1 2 0 1 3 1 2 4 4 3 4 2 1 5 2

Задача I. Грязь

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

— Здравствуйте! Могу я поговорить с Петровым? Алё, милый, привет. . . ты знаешь, у нас дома небольшая авария произошла. . . Но твой компьютер не пострадал, не волнуйся. Но теперь там немного грязно. Ну, то есть очень грязно. Но ты не волнуйся, я приготовила тебе твои болотные сапоги, у входа стоят. А грязь я уберу, как будет свободное время. Когда? Ну, наверное, когда в отпуск пойду. А, ну когда вернёмся из Турции. А, ну значит в следующий отпуск, но обязательно уберу. А пока я у мамы поживу. И ты, кстати, тоже можешь. Ну, как хочешь, я же не заставляю. . . Только пока я не убрала, ты там грязь не разводи, сильно сапогами по грязи не шлёпай и когда по чистому ходишь, сапоги снимай и тапочки обувай, я их тоже возле входа поставила, ты их бери с собой, когда идёшь по грязи и переобувай. А когда по чистому идёшь, бери сапоги, там грязь в разных местах. Программисты, как известно, не самые трудолюбивые люди, поэтому убирать грязь не станут. Но переобувать болотные сапоги каждый раз, когда переходишь от грязного пола к чистому и наоборот — удовольствие ниже среднего, уж лучше пройти лишние несколько метров. Чтобы прожить время до следующего отпуска с комфортом, надо срочно выработать способ добираться из одной точки квартиры с минимальным количеством переобуваний по пути, ну а уж среди них, конечно, выбрать самый короткий.

Формат входных данных

В первой строке даны два целых числа M и N — размеры квартиры (в у.е.). $1 \leq N, M \leq 500$. Два целых числа во второй строке — координаты компьютера (в у.е.), а два целых числа в третьей строке — координаты холодильника (тоже в у.е.). Далее идут M строк по N символов в каждой — план квартиры. На плане 1 означает чистое место, 2 — грязное, 0 — стена или непроходимая грязь. Переходить можно только на клетки, имеющие общую вершину с данной, при переходе с чистой на грязную и наоборот надо переобуваться. Холодильник и компьютер находятся не в клетках, помеченных нулём. Левая верхняя клетка плана имеет координаты (1, 1).

Формат выходных данных

Длина кратчайшего пути (количество преодоленных квадратов, включая начальный и конечный) с минимальным количеством переобуваний, и, через пробел, количество переобуваний (переобувание проходит при переходе с грязного на чистое и наоборот). Если пройти к холодильнику невозможно, вывести числа 0 0.

Пример

stdin	stdout
3 7 1 1 3 7 1200121 1212020 1112021	8 4

Задача J. Раскраска в три цвета

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя нарисовал на бумаге n кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

Формат входных данных

Первая строка содержит два целых числа n и m — количество кружков и количество линий, которые нарисовал Петя, соответственно ($1 \leq n \leq 1\,000$, $0 \leq m \leq 20\,000$).

Следующая строка содержит n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — i -й из этих символов означает цвет, в который раскрашен i -й кружок ('R' — красный, 'G' — зеленый, 'B' — синий).

Следующие m строк содержат по два целых числа — пары кружков, соединенных отрезками.

Формат выходных данных

Выведите в выходной файл одну строку, состоящую из n символов из множества $\{\text{'R'}, \text{'G'}, \text{'B'}\}$ — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово `Impossible`.

Примеры

stdin	stdout
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

Задача К. Транзитивное замыкание

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайт

Дан ориентированный граф. Найдите его транзитивное замыкание, то есть для каждой пары вершин a, b определите, есть ли путь из a в b .

Формат входных данных

На первой строке число вершин n ($1 \leq n \leq 1\,000$). Следующие n строк имеют длину n , состоят из нулей и единиц и задают матрицу смежности графа. Единица в i -й строке, j -м столбце обозначает ребро из i в j .

Формат выходных данных

Выведите матрицу смежности транзитивного замыкания данного графа.

Примеры

stdin	stdout
3	011
010	001
001	000
000	

Задача L. Число

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрежала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

<code>stdin</code>	<code>stdout</code>
2 20 004 66	66220004
3	3

Задача М. Красивый граф

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дан неориентированный граф из N вершин. Каждая вершина либо красная, либо чёрная. Назовем граф красивым, если ни из какой красной вершины не достижима никакая другая красная вершина. Требуется добавить в граф максимальное количество рёбер, чтобы он остался красивым.

Формат входных данных

В первой строке входного файла будет записано одно целое число $1 \leq N \leq 50$. Следующие N строк будут содержать по N символов каждая, i -ый символ j -ой строки будет равен 1, если существует ребро между i -й и j -й вершиной, и 0 иначе. Следующая строка будет содержать одно целое число $1 \leq M \leq N$ — количество красных вершин. Затем будут следовать M целых различных чисел — номера красных вершин. Вершины во входном файле нумеруются с нуля. Гарантируется, что исходный граф красивый.

Формат выходных данных

Выходной файл должен содержать единственное число — максимальное количество рёбер, которое можно добавить в граф так, чтобы он остался при этом красивым.

Примеры

stdin	stdout
3 000 000 000 1 0	3
3 000 000 000 2 0 1	1
2 01 10 1 0	0
5 00000 00000 00000 00000 00000 5 0 1 2 3 4	0
5 01000 10100 01010 00100 00000 2 2 4	3