

Задача А. Разрез

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

Дан неориентированный граф. Найдите минимальный разрез между вершинами 1 и n .

Формат входных данных

На первой строке входного файла содержится n ($1 \leq n \leq 100$) — число вершин в графе и m ($0 \leq m \leq 400$) — количество ребер. На следующих m строках входного файла содержится описание ребер. Ребро описывается номерами вершин, которые оно соединяет, и его пропускной способностью (положительное целое число, не превосходящее 10 000 000), при этом никакие две вершины не соединяются более чем одним ребром.

Формат выходных данных

На первой строке выходного файла должны содержаться количество ребер в минимальном разрезе и их суммарная пропускная способность. На следующей строке выведите возрастающую последовательность номеров ребер (ребра нумеруются в том порядке, в каком они были заданы во входном файле).

Пример

<code>cut.in</code>	<code>cut.out</code>
3 3 1 2 3 1 3 5 3 2 7	2 8 1 2

Замечание

В общем случае для времени работы алгоритма Форда–Фалкерсона нет оценки лучше, чем размер максимального потока на число рёбер. Однако в конкретных задачах он часто оказывается заметно быстрее. Но толкать 10^7 раз строго по одной единичке потока не стоит в любом случае.

Задача В. Просто поток

Имя входного файла: `flow.in`
Имя выходного файла: `flow.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 64 мегабайта

Дана система из узлов и труб, по которым может течь вода. Для каждой трубы известна наибольшая скорость, с которой вода может протекать через нее. Известно, что вода течет по трубам таким образом, что за единицу времени в каждый узел (за исключением двух — источника и стока) втекает ровно столько воды, сколько из него вытекает.

Ваша задача — найти наибольшее количество воды, которое за единицу времени может протекать между источником и стоком, а также скорость течения воды по каждой из труб.

Трубы являются двусторонними, то есть вода в них может течь в любом направлении. Между любой парой узлов может быть более одной трубы.

Формат входных данных

В первой строке записано натуральное число N — количество узлов в системе ($2 \leq N \leq 100$). Известно, что источник имеет номер 1, а сток номер N . Во второй строке записано натуральное M ($1 \leq M \leq 5000$) — количество труб в системе. Далее в M строках идет описание труб. Каждая труба задается тройкой целых чисел A_i, B_i, C_i , где A_i, B_i — номера узлов, которые соединяет данная труба ($A_i \neq B_i$), а C_i ($0 \leq C_i \leq 10^4$) — наибольшая допустимая скорость течения воды через данную трубу.

Формат выходных данных

В первой строке выведите наибольшее количество воды, которое протекает между источником и стоком за единицу времени. Далее выведите M строк, в каждой из которых выведите скорость течения воды по соответствующей трубе. Если направление не совпадает с порядком узлов, заданным во входных данных, то выводите скорость со знаком минус. Числа выводите с точностью 10^{-3} .

Пример

flow.in	flow.out
2	4.0000000
2	1.0000000
1 2 1	-3.0000000
2 1 3	

Задача С. Улиточки

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

Две улиточки Маша и Петя сейчас находятся в на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до n и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). В виду соображений гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

Формат входных данных

В первой строке файла записаны четыре целых числа — n , m , a и h (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика).

В следующих m строках записаны пары чисел. Пара чисел (x, y) означает, что есть дорожка с лужайки x до лужайки y (из-за особенностей улиток и местности дорожки односторонние).

Ограничения: $2 \leq n \leq 10^5$, $0 \leq m \leq 10^5$, $s \neq t$.

Формат выходных данных

Если существует решение, то выведите YES и на двух отдельных строчках сначала путь для Машеньки (т.к. дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

Пример

<code>snails.in</code>	<code>snails.out</code>
3 3 1 3	YES
1 2	1 3
1 3	1 2 3
2 3	

Задача D. Ориентировь меня полностью!

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

Вам дан неориентированный граф без петель и кратных рёбер. Ваша задача — ориентировать граф таким образом, чтобы максимальная исходящая степень была бы минимально возможной.

Формат входных данных

В первой строке заданы числа n и m — количество вершин и рёбер в графе ($1 \leq n \leq 25\,000$; $0 \leq m \leq 25\,000$). В следующих m строках даны пары чисел от 1 до n — рёбра графа.

Формат выходных данных

Выведите минимально возможную максимальную степень. Далее выведите m целых чисел от 0 до 1. Если i -е ребро было задано парой чисел a, b , то ноль означает, что оно после ориентации ведёт из a в b , а единица — что из b в a .

Примеры

<code>orient.in</code>	<code>orient.out</code>
4 4 1 2 1 3 4 2 4 3	1 0 1 1 0
5 5 1 2 2 3 3 1 1 4 1 5	1 0 0 0 1 1