

## Содержание

<b>Must have</b>	<b>2</b>
<b>Задача 3А. Улиточки [0.4 sec, 256 mb]</b>	<b>2</b>
<b>Обязательные задачи</b>	<b>3</b>
<b>Задача 3В. Просто поток [1.5 sec, 256 mb]</b>	<b>3</b>
<b>Задача 3С. Разрез [0.2 sec, 256 mb]</b>	<b>4</b>
<b>Задача 3D. Декомпозиция потока [0.5 sec, 256 mb]</b>	<b>5</b>
<b>Задача 3Е. Molecule. Химия!!! [0.3 sec, 256 mb]</b>	<b>6</b>
<b>Дополнительные задачи</b>	<b>7</b>
<b>Задача 3F. Великая стена [0.8 sec, 256 mb]</b>	<b>7</b>
<b>Задача 3G. Матан [0.3 sec, 256 mb]</b>	<b>8</b>
<b>Задача 3H. Hard life [0.3 sec, 256 mb]</b>	<b>9</b>

---

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же `stdin`), вывести ответ нужно в **стандартный поток вывода** (он же `stdout`).

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

## Must have

### Задача 3А. Улиточки [0.4 sec, 256 mb]

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 —  $n$ ,  $m$ ,  $s$  и  $t$  (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 последовательность лужаек для Машеньки (дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

#### Пример

stdin	stdout
3 3 1 3	YES
1 2	1 3
1 3	1 2 3
2 3	

#### Замечание

Дан оргграф, найти два непересекающихся по ребрам пути из  $s$  в  $t$ , вывести вершины найденных путей.

## Обязательные задачи

### Задача 3В. Просто поток [1.5 sec, 256 mb]

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

stdin	stdout
2	4.0000000
2	1.0000000
1 2 1	-3.0000000
2 1 3	

#### Замечание

Должно заходить что угодно, толкающее  $\min(c_e - f_e)$ .

Будьте бдительны! Граф **неориентирован**.

### Задача 3С. Разрез [0.2 sec, 256 mb]

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

stdin	stdout
3 3	2 8
1 2 3	1 2
1 3 5	
3 2 7	

### Задача 3D. Декомпозиция потока [0.5 сек, 256 mb]

Задан ориентированный граф, каждое ребро которого обладает целочисленной пропускной способностью. Найдите максимальный поток из вершины с номером 1 в вершину с номером  $n$  и постройте декомпозицию этого потока.

#### Формат входных данных

Первая строка входного файла содержит  $n$  и  $m$  — количество вершин и количество ребер графа ( $2 \leq n \leq 500$ ,  $1 \leq m \leq 10\,000$ ). Следующие  $m$  строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят  $10^9$ .

#### Формат выходных данных

В первую строку выходного файла выведите одно число — количество путей в декомпозиции максимального потока из вершины с номером 1 в вершину с номером  $n$ . Следующий строки должны содержать описания элементарных потоков, на который был разбит максимальный. Описание следует выводить в следующем формате: величина потока, количество ребер в пути, вдоль которого течет данный поток и номера ребер в этом пути. Ребра нужно выводить в порядке прохождения. Ребра нумеруются с единицы в порядке появления во входном файле.

#### Примеры

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

#### Замечание

Должно заходить масштабирование потока, толкающее по пути  $\min_e(c_e - f_e)$ .  
Будьте бдительны! Граф **ориентирован**.

### Задача 3Е. Molecule. Химия!!! [0.3 sec, 256 mb]

Вася и Сережа играют в следующую игру. В некоторых клетках клетчатого листка Сережа рисует один из символов 'H', 'O', 'N' или 'C', после чего Вася должен провести между некоторыми находящимися в соседних клетках символами линии так, чтобы получилось корректное изображение химической молекулы. К сожалению, Сережа любит рисовать много символов, и Вася не может сразу определить, возможно ли вообще нарисовать линии нужным способом. Помогите ему написать программу, которая даст ответ на этот вопрос.

В этой задаче проведенные между символами химических элементов линии будем считать корректным изображением молекулы, если они удовлетворяют следующим условиям:

- каждая линия соединяет символы, нарисованные в соседних (по стороне) клетках,
- между каждой парой символов проведено не более одной линии,
- от каждого элемента отходит ровно столько линий, какова валентность этого элемента (1 для H, 2 для O, 3 для N, 4 для C),
- пустые клетки ни с чем не соединены, и
- **хотя бы в одной клетке** нарисован какой-то символ.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  ( $1 \leq n, m \leq 50$ ) — размеры листочка, на котором рисует Сережа. Далее следуют  $n$  строк по  $m$  символов в каждой, задающих конфигурацию химических элементов, которую нарисовал Сережа; пустые клетки задаются символом '.'.

#### Формат выходных данных

В выходной файл выведите одно слово: 'Valid', если линии провести требуемым образом можно, и 'Invalid', если нельзя.

#### Пример

stdin	stdout
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid

## Дополнительные задачи

### Задача 3F. Великая стена [0.8 сек, 256 mb]

У короля Людовика двое сыновей. Они ненавидят друг друга, и король боится, что после его смерти страна будет уничтожена страшными войнами. Поэтому Людовик решил разделить свою страну на две части, в каждой из которых будет властвовать один из его сыновей. Он посадил их на трон в города  $A$  и  $B$ , и хочет построить минимально возможное количество фрагментов стены таким образом, чтобы не существовало пути из города  $A$  в город  $B$ .

Страну, в которой властвует Людовик, можно упрощенно представить в виде прямоугольника  $m \times n$ . В некоторых клетках этого прямоугольника расположены горы, по остальным же можно свободно перемещаться. Кроме этого, ландшафт в некоторых клетках удобен для строительства стены, в остальных же строительство невозможно.

При поездках по стране можно перемещаться из клетки в соседнюю по стороне, только если ни одна из этих клеток не содержит горы или построенного фрагмента стены.

#### Формат входных данных

В первой строке входного файла содержатся числа  $m$  и  $n$  ( $1 \leq m, n \leq 50$ ). Во второй строке заданы числа  $k$  и  $l$ , где  $0 \leq k, l, k + l \leq mn - 2$ ,  $k$  — количество клеток, на которых расположены горы, а  $l$  — количество клеток, на которых можно строить стену. Естественно, что на горах строить стену нельзя. Следующие  $k$  строк содержат координаты клеток с горами  $x_i$  и  $y_i$ , а за ними следуют  $l$  строк, содержащие координаты клеток, на которых можно построить стену —  $x_j$  и  $y_j$ . Последние две строки содержат координаты городов  $A$  ( $x_A$  и  $y_A$ ) и  $B$  ( $x_B$  и  $y_B$ ) соответственно. Среди клеток, описанных в этих  $k + l + 2$  строках, нет двух совпадающих. Гарантируется, что  $1 \leq x_i, x_j, x_A, x_B \leq m$  и  $1 \leq y_i, y_j, y_A, y_B \leq n$ .

#### Формат выходных данных

В первой строке выходного файла должно быть выведено минимальное количество фрагментов стены  $F$ , которые необходимо построить. В последующих  $F$  строках необходимо вывести один из возможных вариантов застройки.

Если невозможно произвести требуемую застройку, то необходимо вывести в выходной файл единственное число  $-1$ .

#### Пример

stdin	stdout
5 5	3
3 8	3 1
3 2	1 3
2 4	3 3
3 4	
3 1	
1 3	
2 3	
3 3	
4 3	
5 3	
1 4	
1 5	
2 1	
5 5	

### Задача 3G. Матан [0.3 sec, 256 mb]

В Университете города М. проводят эксперимент. Преподаватели сами решают, что они будут читать в рамках того или иного курса. И вот преподаватель математического анализа (в простонародье — матана) оценил по некоторым критериям все известные ему темы в данном курсе. В результате этой ревизии каждой теме сопоставлено некоторое целое число (возможно, отрицательное) — полезность данной темы. Профессор хочет максимизировать суммарную полезность прочитанных им тем, но не все так просто. Для того что бы студенты поняли некоторые темы, необходимо, чтобы были прочитаны так же некоторые другие темы, так как некоторые доказательства базируются на фактах из других тем. Однако если существует цикл из зависимостей тем, то их все можно прочитать, и на качестве понимания материала студентами это не скажется.

Вас попросили составить список тем, которые профессор должен прочитать, таким образом, чтобы студенты все поняли, и суммарная полезность курса была максимальна.

#### Формат входных данных

Первая строка входного файла содержит одно число —  $N$  ( $1 \leq N \leq 200$ ). Вторая строка содержит  $N$  целых чисел, не превосходящих по модулю 1 000 — полезности каждой темы. Далее следуют  $N$  строк с описанием зависимостей тем. Каждое описание начинается количеством тем, которые необходимо понять для понимания данной темы. Потом следуют номера этих тем, разделенные пробелами. Суммарное количество рёбер не более 1 800.

#### Формат выходных данных

Выведите единственное число — максимально возможную суммарную полезность прочитанного материала.

#### Примеры

stdin	stdout
4 -1 1 -2 2 0 1 1 2 4 2 1 1	2
3 2 -1 -2 2 2 3 0 0	0

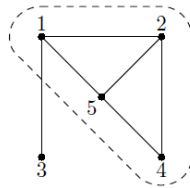
#### Замечание

Разбиралась на практике...



### Задача 3Н. Hard life [0.3 sec, 256 mb]

John is a Chief Executive Officer at a privately owned medium size company. The owner of the company has decided to make his son Scott a manager in the company. John fears that the owner will ultimately give CEO position to Scott if he does well on his new manager position, so he decided to make Scott's life as hard as possible by carefully selecting the team he is going to manage in the company. John knows which pairs of his people work poorly in the same team. John introduced a hardness factor of a team — it is a number of pairs of people from this team who work poorly in the same team divided by the total number of people in the team. The larger is the hardness factor, the harder is this team to manage. John wants to find a group of people in the company that are harderst to manage and make it Scott's team. Please, help him.



In the example on the picture the hardest team consists of people 1, 2, 4, and 5. Among 4 of them 5 pairs work poorly in the same team, thus hardness factor is equal to  $\frac{5}{4}$ . If we add person number 3 to the team then hardness factor decreases to  $\frac{6}{5}$ .

#### Формат входных данных

The first line of the input file contains two integer numbers  $n$  and  $m$  ( $1 \leq n \leq 100, 0 \leq m \leq 1000$ ). Here  $n$  is a total number of people in the company (people are numbered from 1 to  $n$ ), and  $m$  is the number of pairs of people who work poorly in the same team. Next  $m$  lines describe those pairs with two integer numbers  $a_i$  and  $b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ) on a line. The order of people in a pair is arbitrary and no pair is listed twice.

#### Формат выходных данных

Write to the output file an integer number  $k$  ( $1 \leq k \leq n$ ) — the number of people in the hardest team, followed by  $k$  lines listing people from this team in ascending order. If there are multiple teams with the same hardness factor then write any one.

#### Пример

stdin	stdout
5 6	4
1 5	1
5 4	2
4 2	4
2 5	5
1 2	
3 1	
4 0	1
	1

#### Замечание

Найти подграф максимальной средней степени.