

## Содержание

<b>09.Base [2.2]</b>	<b>3</b>
Задача 9A. Просто поток [0.5 sec (1 sec), 256 mb]	3
Задача 9B. Разрез [0.5 sec (1 sec), 256 mb]	4
<b>09.Advanced [3/5]</b>	<b>5</b>
Задача 9C. Улиточки [0.5 sec (1 sec), 256 mb]	5
Задача 9D. Максимальный поток [0.5 sec (1 sec), 256 mb]	6
Задача 9E. Molecule. Химия!!! [0.5 sec (1 sec), 256 mb]	7
Задача 9F. Великая стена [0.5 sec (1 sec), 256 mb]	8
Задача 9G. Контролирующее множество [0.5 sec (1 sec), 256 mb]	9
<b>09.Hard [0/3]</b>	<b>10</b>
Задача 9H. Охлаждение реактора [0.5 sec (1 sec), 256 mb]	10
Задача 9I. Живопись [0.5 sec (1 sec), 256 mb]	12
Задача 9J. Матан [0.5 sec (1 sec), 256 mb]	13

**Общая информация:**

Вход в констест: <http://contest.yandex.ru/contest/1183/>

Дедлайн на задачи: 2 недели, до 26-го апреля 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <https://compscicenter.ru/courses/algorithms-2/2015-spring/>

Семинары ведёт Сергей Владимирович Копелиович,  
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии указан таймлит для C/C++.

Таймлит для Java примерно в 2-3 раза больше.

Таймлит для Python примерно в 4-5 раз больше.

## 09.Base [2.2]

### Задача 9А. Просто поток [0.5 sec (1 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

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

### Задача 9B. Разрез [0.5 sec (1 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

cut.in	cut.out
3 3	2 8
1 2 3	1 2
1 3 5	
3 2 7	

## 09.Advanced [3/5]

### Задача 9C. Улиточки [0.5 sec (1 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

snails.in	snails.out
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.

### Задача 9D. Максимальный поток [0.5 sec (1 sec), 256 mb]

Вам задан ориентированный граф  $G$ . Каждое ребро имеет некоторую пропускную способность. Найдите максимальный поток между вершинами 1 и  $n$ .

#### Формат входных данных

Первая строка входного файла содержит  $n$  и  $m$  — число вершин и ребер в графе ( $2 \leq n \leq 500$ ,  $1 \leq m \leq 10\,000$ ). Последующие строки описывают ребра. Каждое ребро задается тремя числами: начальная вершина ребра, конечная вершина ребра и пропускная способность ребра. Пропускные способности не превосходят  $10^9$ .

#### Формат выходных данных

Выведите величину максимального потока между вершинами 1 и  $n$ .

Далее для каждого ребра выведите величину потока, текущую по этому ребру.

#### Примеры

flow2.in	flow2.out
4 5	3.0
1 2 1	1.0
1 3 2	2.0
3 2 1	1.0
2 4 2	2.0
3 4 1	1.0

### Задача 9Е. Molecule. Химия!!! [0.5 sec (1 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

molecule.in	molecule.out
3 4 HOH. NCOH OO..	Valid
3 4 HOH. NCOH OONH	Invalid

### Задача 9F. Великая стена [0.5 sec (1 sec), 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

wall.in	wall.out
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 9G. Контролирующее множество [0.5 sec (1 sec), 256 mb]

Дан взвешенный двудольный граф. Требуется найти контролирующее множество минимального веса. Контролирующим множеством называется множество вершин такое, что у каждого ребра выбран хотя бы один из двух концов. Весом контролирующего множества называется суммарный вес вершин множества.

#### Формат входных данных

В первой строке дано целое число  $T$  — количество тестов.

Далее идут  $T$  тестов.

$N$  ( $1 \leq N \leq 100$ ) — количество вершин в первой доле. В следующей строке записаны веса вершин в первой доле.

Далее идет число  $M$  ( $1 \leq M \leq 100$ ) — количество вершин во второй доле. В следующей строке записаны веса вершин во второй доле.

Затем дано число  $E$  ( $0 \leq E \leq 10^4$ ) — количество ребер в графе. В каждой из последующих  $E$  строк содержится два числа  $u, v$  ( $0 \leq u \leq N - 1, 0 \leq v \leq M - 1$ ), которые описывают ребро.

Суммарное количество  $N, M$  и  $E$  не превосходит 30 000.

Веса вершин неотрицательное число, которое не превосходит  $10^9$ .

#### Формат выходных данных

Для каждого теста выведете числа  $n$  и  $m$  количество вершин в контролирующем множестве в первой и во второй доле соответственно. На следующей строке вершины из первой доли, которые входят в множество. На следующей строке вершины из второй доли, которые входят в множество.

Если оптимальных ответов несколько, выведите любой из них.

#### Примеры

controls.in	controls.out

## 09.Hard [0/3]

### Задача 9H. Охлаждение реактора [0.5 sec (1 sec), 256 mb]

Известная террористическая группа под руководством знаменитого террориста Бен Гадена решила построить атомный реактор для получения оружейного плутония. Вам, как компьютерному гению этой группы, поручили разработать систему охлаждения реактора.

Система охлаждения реактора представляет собой набор труб, соединяющих узлы. По трубам течет жидкость, причем для каждой трубы строго определено направление, в котором она должна по ней течь. Узлы системы охлаждения занумерованы от 1 до  $N$ . Система охлаждения должна быть спроектирована таким образом, чтобы для каждого узла за единицу времени количество жидкости, втекающей в узел, было равно количеству жидкости, вытекающей из узла. То есть если из  $i$ -го узла в  $j$ -ый течет  $f_{ij}$  единиц жидкости за единицу времени (если из  $i$  в  $j$  нет трубы, то положим  $f_{ij} = 0$ ), то для каждого узла  $i$  должно выполняться

$$\sum_{j=1}^N f_{ij} = \sum_{j=1}^N f_{ji}$$

У каждой трубы имеется пропускная способность  $c_{ij}$ . Кроме того, для обеспечения достаточного охлаждения требуется, чтобы по трубе протекало не менее  $l_{ij}$  единиц жидкости за единицу времени. То есть для трубы, ведущей из  $i$ -го узла в  $j$ -ый должно выполняться  $l_{ij} \leq f_{ij} \leq c_{ij}$ .

Вам дано описание системы охлаждения, выясните, каким образом можно пустить жидкость по трубам, чтобы выполнялись все указанные условия.

#### Формат входных данных

Первая строка входного файла содержит числа  $N$  и  $M$  — количество узлов и труб ( $1 \leq N \leq 200$ ). Следующие  $M$  строк содержат описание труб. Каждая строка содержит четыре целых числа  $i$ ,  $j$ ,  $l_{ij}$  и  $c_{ij}$ . Любые два узла соединены не более чем одной трубой, если есть труба из  $i$  в  $j$ , то нет трубы из  $j$  в  $i$ , никакой узел не соединен трубой сам с собой,  $0 \leq l_{ij} \leq c_{ij} \leq 10^5$ .

#### Формат выходных данных

Если решение существует, выведите на первой строке выходного файла слово YES. Затем выведите  $M$  чисел — количество жидкости, которое должно течь по трубам, числа должны быть выведены в том порядке, в котором трубы заданы во входном файле. Если решения не существует, выведите NO.

### Пример

cooling.in	cooling.out
4 6 1 2 1 2 2 3 1 2 3 4 1 2 4 1 1 2 1 3 1 2 4 2 1 2	NO
4 6 1 2 1 3 2 3 1 3 3 4 1 3 4 1 1 3 1 3 1 3 4 2 1 3	YES 1 2 3 2 1 1

### Задача 9I. Живопись [0.5 sec (1 sec), 256 mb]

В стране Олимпия очень развита живопись. Картиной считается любой прямоугольник, который состоит из черных и белых единичных квадратов. Художник Олимпус решил радикально улучшить свои картины. Для этого он планирует к белому и черному цветам добавить еще и серый оттенок. По его задумке, граница между каждым черным и белым квадратом должна содержать серую линию, чтобы образовался эффект плавного перехода.

Однако, перед началом работы, он обнаружил, что серая краска очень дорого стоит. Чтобы сэкономить деньги художник решил оценить, не выгоднее ли сначала перекрасить некоторые белые квадраты в черные, а черные в белые для того, чтобы минимизировать расходы на краску.

Напишите программу, которая по информации о существующей картине определяет минимальную сумму денег, которые понадобятся на ее улучшение.

#### Формат входных данных

Первая строка входного файла содержит пять натуральных чисел  $N$ ,  $M$ ,  $w$ ,  $b$ ,  $g$ .  $1 \leq N, M \leq 70$  — высота и ширина картины,  $1 \leq w, b, g \leq 1000$  — цена рисования одного белого единичного квадрата, черного единичного квадрата и серой линии единичной длины, соответственно. Далее следует  $N$  строк, каждая из которых состоит из  $M$  литер. Литера  $B$  соответствует черному квадрату, а  $W$  — белому.

#### Формат выходных данных

Единственная строка выходного файла должна содержать одно целое число, которое есть минимальной суммой затрат на улучшение картины.

#### Пример

drawing.in	drawing.out
3 2 3 3 2 BB WW WB	7

### Задача 9J. Матан [0.5 сек (1 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 прочесть, и на качестве понимания материала студентами это не скажется.

Вас попросили составить список тем, которые профессор должен прочесть, 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

matan.in	matan.out
4 -1 1 -2 2 0 1 1 2 4 2 1 1	2
3 2 -1 -2 2 2 3 0 0	0