

## Содержание

<b>Обязательные задачи</b>	<b>3</b>
Задача 24А. Максимальный поток минимальной стоимости [0.3 sec, 256 mb]	3
Задача 24В. В поисках невест [0.3 sec, 256 mb]	4
Задача 24С. Задача о назначениях [1.5 sec, 256 mb]	5
Задача 24D. $k$ паросочетаний [0.5 sec, 256 mb]	6
Задача 24Е. Максимальное паросочетание [1.5 sec, 256 mb]	7
Задача 24F. Толстые хоббиты [0.3 sec, 256 mb]	8
Задача 24G. Охлаждение реактора [0.3 sec, 256 mb]	9
<b>Бонус</b>	<b>11</b>
Задача 24H. Автоматное программирование [0.25 sec, 256 mb]	11
Задача 24I. Живопись [1 sec, 256 mb]	12
Задача 24J. Sociology [0.5 sec, 256 mb]	13
Задача 24K. Кредитные операции 2 [0.5 sec, 256 mb]	14
Задача 24L. Глобальный разрез [1.5 sec, 256 mb]	15

---

### Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

### Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

[http://acm.math.spbu.ru/~sk1/algo/input-output/cpp\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html)

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

[http://acm.math.spbu.ru/~sk1/algo/input-output/io\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html)

[http://acm.math.spbu.ru/~sk1/algo/input-output/fread\\_write\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html)

### Выделение памяти.

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

## Обязательные задачи

### Задача 24А. Максимальный поток минимальной стоимости [0.3 sec, 256 mb]

Задан ориентированный граф, каждое ребро которого обладает пропускной способностью и стоимостью. Найдите максимальный поток минимальной стоимости из вершины с номером 1 в вершину с номером  $n$ .

#### Формат входных данных

Первая строка входного файла содержит  $n$  и  $m$  — количество вершин и количество ребер графа ( $2 \leq n \leq 100$ ,  $0 \leq m \leq 1000$ ). Следующие  $m$  строк содержат по четыре целых числа: номера вершин, которые соединяет соответствующее ребро графа, его пропускную способность и его стоимость. Пропускные способности и стоимости не превосходят  $10^5$ .

#### Формат выходных данных

В выходной файл выведите одно число — цену максимального потока минимальной стоимости из вершины с номером 1 в вершину с номером  $n$ . Ответ не превышает  $2^{63} - 1$ . Гарантируется, что в графе нет циклов отрицательной стоимости.

#### Примеры

mincost.in	mincost.out
4 5 1 2 1 2 1 3 2 2 3 2 1 1 2 4 2 1 3 4 2 3	12

#### Подсказка по решению

В этой задаче достаточно несколько раз пустить Форд-Беллмана...

### Задача 24В. В поисках невест [0.3 sec, 256 mb]

Однажды король Флатландии решил отправить  $k$  своих сыновей на поиски невест. Всем известно, что во Флатландии  $n$  городов, некоторые из которых соединены дорогами. Король живет в столице, которая имеет номер 1, а город с номером  $n$  знаменит своими невестами.

Итак, король повелел, чтобы каждый из его сыновей добрался по дорогам из города 1 в город  $n$ . Поскольку, несмотря на обилие невест в городе  $n$ , красивых среди них не так много, сыновья опасаются друг друга. Поэтому они хотят добраться до цели таким образом, чтобы никакие два сына не проходили по одной и той же дороге (даже в разное время). Так как король любит своих сыновей, он хочет, чтобы среднее время сына в пути до города назначения было минимально.

#### Формат входных данных

В первой строке входного файла находятся числа  $n$ ,  $m$  и  $k$  — количество городов и дорог во Флатландии и сыновей короля, соответственно ( $2 \leq n \leq 200$ ,  $1 \leq m \leq 2000$ ,  $1 \leq k \leq 100$ ). Следующие  $m$  строк содержат по три целых положительных числа каждая — города, которые соединяет соответствующая дорога и время, которое требуется для ее прохождения (время не превышает  $10^6$ ). По дороге можно перемещаться в любом из двух направлений, два города могут быть соединены несколькими дорогами.

#### Формат выходных данных

Если выполнить повеление короля невозможно, выведите на первой строке число  $-1$ . В противном случае выведите на первой строке минимальное возможное среднее время (с точностью 5 знаков после десятичной точки), которое требуется сыновьям, чтобы добраться до города назначения, не менее чем с пятью знаками после десятичной точки. В следующих  $k$  строках выведите пути сыновей, сначала число дорог в пути, и затем номера дорог в пути в том порядке, в котором их следует проходить. Дороги нумеруются, начиная с единицы, в том порядке, в котором они заданы во входном файле.

#### Пример

brides.in	brides.out
5 8 2	3.00000
1 2 1	3 1 5 6
1 3 1	3 2 7 8
1 4 3	
2 5 5	
2 3 1	
3 5 1	
3 4 1	
5 4 1	

#### Замечание

Вы же писали поток dfs-ом? Нужно вместо dfs-а искать путь Форд-Беллманом. Долго работает? Вы умеете заставить Форд-Беллманом работать быстро.

**Задача 24С. Задача о назначениях [1.5 сек, 256 mb]**

Дана целочисленная матрица  $C$  размера  $n \times n$ . Требуется выбрать  $n$  ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка и сумма значений в выбранных ячейках было минимальна.

**Формат входных данных**

Первая строка входного файла содержит  $n$  ( $2 \leq n \leq 300$ ). Каждая из последующих  $n$  строк содержит по  $n$  чисел:  $C_{ij}$ . Все значения во входном файле неотрицательны и не превосходят  $10^6$ .

**Формат выходных данных**

В первую строку выходного файла выведите одно число — искомая минимизируемая величина. Далее выведите  $n$  строк по два числа в каждой — номер строки и столбца клетки, участвующей в оптимальном назначении.

Пары чисел можно выводить в произвольном порядке.

**Примеры**

assignment.in	assignment.out
3	3
3 2 1	2 1
1 3 2	3 2
2 1 3	1 3

**Замечание**

Можно сдать Венгреку, можно поток с Дейкстрой.

**Задача 24D.  $k$  паросочетаний [0.5 sec, 256 mb]**

Дан полный взвешенный двудольный граф с равным количеством вершин в долях. Требуется выбрать  $k$  максимальных попарно не пересекающихся паросочетаний так, чтобы их суммарный вес был минимален.

**Формат входных данных**

Первая строка входного файла содержит  $n$  и  $k$  — количество вершин в каждой из долей и количество паросочетаний ( $2 \leq n \leq 50$ ,  $1 \leq k \leq n$ ). Каждая из последующих  $n$  строк содержит по  $n$  чисел:  $C_{ij}$  — вес ребра, ведущего из  $i$ -й вершины левой доли в  $j$ -ю правой.

Все значения во входном файле неотрицательны и не превосходят  $10^6$ .

**Формат выходных данных**

В первую строку выходного файла выведите одно число — искомый суммарный вес паросочетаний. Следующие  $k$  строк должны содержать  $n$  чисел — номера вершины, правой доли, соответствующие вершинам левой.

**Примеры**

multiassignment.in	multiassignment.out
3 2	6
1 2 1	1 2 3
1 1 2	3 1 2
2 1 1	

### Задача 24Е. Максимальное паросочетание [1.5 sec, 256 mb]

Дан двудольный граф. У каждой вершины графа есть вес. Вес ребра — сумма весов его концов. Вес паросочетания — сумма весов рёбер, входящих в паросочетание. Нужно найти паросочетание максимального веса. Заметим, это паросочетание может содержать сколько угодно рёбер, единственное условие — вес паросочетания должен быть максимальным.

Напомним, что паросочетанием в двудольном графе называется набор рёбер этого графа такой, что никакие два ребра набора не имеют общих вершин.

#### Формат входных данных

В первой строке заданы размеры долей  $n$  и  $m$  ( $1 \leq n, m \leq 5000$ ) и количество рёбер  $e$  ( $0 \leq e \leq 10000$ ). Вторая строка содержит  $n$  целых чисел от 0 до 10000 — веса вершин первой доли. Третья строка содержит  $m$  целых чисел от 0 до 10000 — веса вершин второй доли. Следующие  $e$  строк содержат рёбра графа. Каждое ребро описывается парой целых чисел  $a_i b_i$ , где  $1 \leq a_i \leq n$  — номер вершины первой доли и  $1 \leq b_i \leq m$  — номер вершины второй доли.

#### Формат выходных данных

В первой строке выведите  $w$  — максимальный вес паросочетания. Во второй строке выведите  $k$  — количество рёбер в паросочетании максимального веса. В следующей строке выведите  $k$  различных чисел от 1 до  $e$  — номера рёбер в паросочетании. Если максимальных по весу паросочетаний несколько, разрешается вывести одно любое.

#### Примеры

matching.in	matching.out
4 3 3 2 0 9 9 1 0 9 1 2 2 1 1 1	3 1 3
3 2 4 1 2 3 1 2 1 1 2 1 2 2 3 2	8 2 4 2

#### Замечание

Эта задача два раза была на практике. Если вы умеете писать dfs и таки запомнили, как её решать, вот вам бесплатный плюсики.

### Задача 24F. Толстые хоббиты [0.3 sec, 256 mb]

Ни один хоббит не в состоянии в одиночку противостоять полчищам Мордора... В последний поход против Мордора Гэндальф решил отправить  $N$  хоббитов из Шира. Но часть хоббитов наотрез отказалась, жалуясь на то, что другие хоббиты наверняка будут дразнить их толстыми. После опроса всех хоббитов оказалось, что любой хоббит отказывается принять участие в походе в том случае, если с ним в поход выступит хотя бы один хоббит с меньшим весом. К счастью для Средиземья, не все хоббиты знают свой точный вес. В Шире были всего одни весы чашечного типа, позволяющие для пары хоббитов определить, какой хоббит тяжелее. Некоторые пары хоббитов взвешивались на этих весах. Всем хоббитам известен результат всех взвешиваний. Гэндальф абсолютно уверен, что в Шире нет двух хоббитов одного веса. Он заинтересован в том, чтобы отряд состоял из наибольшего количества хоббитов. Однако найти наибольшее множество хоббитов, среди которых ни один не считает себя тяжелее другого, оказалось не так-то просто. Подскажите Гэндальфу, на сколько хоббитов он может рассчитывать. Помните при этом, что хоббиты умные существа и знают, что если Сэм тяжелее Пиппина, а Пиппин тяжелее Фродо, то Сэм и подавно будет тяжелее Фродо.

#### Формат входных данных

В первой строке дано целое число  $N$  — количество хоббитов ( $2 \leq N \leq 100$ ). Все хоббиты пронумерованы целыми числами от 1 до  $N$ . В следующих  $N$  строках записана матрица размера  $N \times N$ . Если  $i$ -й и  $j$ -й хоббит взвешивались на чашечных весах и оказалось, что  $i$ -й хоббит тяжелее, то в  $i$ -й строке матрицы на  $j$ -й позиции стоит единица. Во всех остальных случаях в матрице стоят нули.

#### Формат выходных данных

В первой строке выведите размер наибольшего множества хоббитов, готового выступить в поход, во второй строке перечислите номера хоббитов из этого множества через пробел.

#### Пример

hobbits.in	hobbits.out
2 0 1 0 0	1 2
3 0 0 0 0 0 0 0 0 0	3 1 2 3

#### Замечание

Вы же помните теорему Дилворта? Была задача на практике.



### Задача 24G. Охлаждение реактора [0.3 сек, 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

## Бонус

### Задача 24Н. Автоматное программирование [0.25 sec, 256 mb]

В один замечательный день в компанию «X» завезли  $k$  автоматов. И не простых автоматов, а автоматов-программистов! Это был последний неудачный шаг перед переходом на андроидов-программистов, но это уже совсем другая история.

В компании сейчас  $n$  задач, для каждой из которых известно время начала ее выполнения  $s_i$ , длительность ее выполнения  $t_i$  и прибыль компании от ее завершения  $c_i$ . Любой автомат может выполнять любую задачу, ровно одну в один момент времени. Если автомат начал выполнять задачу, то он занят все моменты времени с  $s_i$  по  $s_i + t_i - 1$  включительно и не может переключиться на другую задачу.

Вам требуется выбрать набор задач, которые можно выполнить с помощью этих  $k$  автоматов и который принесет максимальную суммарную прибыль.

#### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $k$  ( $1 \leq n \leq 1000$ ,  $1 \leq k \leq 50$ ) — количество задач и количество автоматов, соответственно.

В следующих  $n$  строках через пробелы записаны тройки целых чисел  $s_i, t_i, c_i$  ( $1 \leq s_i, t_i \leq 10^9$ ,  $1 \leq c_i \leq 10^6$ ),  $s_i$  — время начала выполнения  $i$ -го задания,  $t_i$  — длительность  $i$ -го задания, а  $c_i$  — прибыль от его выполнения.

#### Формат выходных данных

Выведите  $n$  целых чисел  $x_1, x_2, \dots, x_n$ . Число  $x_i$  должно быть равно 1, если задачу  $i$  следует выполнить, и 0 в противном случае.

Если оптимальных решений несколько, то выведите любое из них.

#### Примеры

schedule.in	schedule.out
3 1 2 7 5 1 3 3 4 1 3	0 1 1
5 2 1 5 4 1 4 5 1 3 2 4 1 2 5 6 1	1 1 0 0 1

#### Замечание

В первом примере задания требуют выполнения в моменты времени 2...8, 1...3 и 4...4, соответственно. Первое задание пересекается со вторым и третьим, поэтому можно выполнять либо его одно (прибыль 5), либо второе и третье (прибыль 6).

#### Подсказка по решению

В этой задаче нужно построить граф из  $\mathcal{O}(n)$  вершин и рёбер.

### Задача 24I. Живопись [1 сек, 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 24J. Sociology [0.5 sec, 256 mb]

Vasya works for the RISK (Research Institute of Sociological tasKs). He is studying relationships between software engineers working freelance and their managers. Vasya considers several jobs assigned to programmers. Each job is to be done by one engineer and is to be managed by one manager.

Vasya calls a subset  $A$  of managers *excessive* if the subset of engineers having common jobs with at least one of managers from  $A$  has lesser cardinality than  $|A|$ . His hypothesis is that a system having no excessive subsets of managers is more stable and produces less pressure to the workers.

Your task is to find an excessive subset or say that it is impossible.

#### Формат входных данных

Input consists of no more than 10 test cases. The first line of each test case contains two integers  $N_e$  and  $N_m$  — the number of engineers and managers, respectively ( $1 \leq N_e, N_m \leq 10^4$ ). The next line contains a single integer  $N_j$  — the number of jobs ( $1 \leq N_j \leq 10^5$ ). Then  $N_j$  lines follow, each containing two integers  $e_i$  and  $m_i$  — numbers of engineer and manager assigned to job number  $i$ .

#### Формат выходных данных

For each test case, output either an excessive subset of managers or a message that it does not exist. Adhere to the sample output below as close as possible.

#### Пример

sociology.in	
3	3
6	
1	2
1	3
2	3
2	1
3	1
3	2
2	3
2	
1	3
2	2
1	3
3	
1	1
1	2
1	3

  

sociology.out	
Case #1:	There is no excessive subset of managers.
Case #2:	Manager 1 forms an excessive subset.
Case #3:	Managers 1, 2 and 3 form an excessive subset.

#### Замечание

Условие: найти контрпример к лемме Холла.

Решение: по замыслу должен заходить только Хопкрофт-Карп.

**Задача 24К. Кредитные операции 2 [0.5 сек, 256 mb]**

Дана квадратная матрица  $a_{ij}$  из неотрицательных целых чисел. Найти такие целые неотрицательные  $x_i, y_j: \forall i, j x_i + y_j \geq a_{ij}$  и при этом  $\sum x_i + \sum y_j$  минимальна. В записи  $a_{ij}$ :  $i$  – номер строки ячейки,  $j$  – номер столбца ячейки.

**Формат входных данных**

На первой строке число  $n$  ( $1 \leq n \leq 100$ ).

Следующие  $n$  строк содержат матрицу  $n \times n$  из целых чисел от 0 до  $10^6$ .

**Формат выходных данных**

На первой строке вектор  $x$ . На второй строке вектор  $y$ .

**Примеры**

credit.in	credit.out
4	2 0 1 2
5 8 4 3	3 6 3 2
3 6 2 1	
4 6 4 1	
4 3 5 4	

**Замечание**

Привет, Венгерка.

### Задача 24L. Глобальный разрез [1.5 сек, 256 mb]

Дан взвешенный неориентированный граф. Нужно удалить минимальное по сумме весов множество ребер так, чтобы граф стал несвязным. После того, как граф становится несвязным, множество вершин распадается на два непустых множества  $S$  и  $T$ . Зная одно из двух множеств, можно восстановить второе и удаленные ребра.

#### Формат входных данных

Количество вершин и ребер в графе —  $2 \leq N \leq 500, 1 \leq M \leq 125\,000$ .

Далее  $M$  строк, в каждой 3 числа —  $1 \leq a, b \leq N$  и  $0 \leq w \leq 10^9$  (вершины, которые соединяет очередное ребро, и вес ребра).

#### Формат выходных данных

Минимальный суммарный вес ребер, которых нужно удалить.

Далее размер множества  $S$  и собственно вершины из  $S$ .

#### Пример

stor.in	stor.out
3 3	3
1 2 2	2
2 3 3	2 3
3 1 1	
6 7	1
1 2 10	3
2 3 10	4 5 6
3 1 10	
4 5 10	
5 6 10	
6 4 10	
2 5 1	

#### Замечание

Можно сдать Штор-Вагнера, можно Каргера-Штейна.