

# ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Решение задач этого раздела предполагает использование известных и разработку новых алгоритмов на графах. Базовые алгоритмы не следует как-то модифицировать, вместо этого лучше грамотно построить граф-модель задачи, чтобы использовать их без изменений. Приемлемое время работы алгоритма определяется исходя из ограничений на входные данные в каждой задаче. Полученная для алгоритма оценка времени работы должна гарантировать, что решение уложится в отведённое время на любом наборе входных данных, соответствующем ограничениям, указанным в условии задачи.

Можно использовать любой язык программирования.

## Задача 1. Наибольшее число маршрутов (не пересекающихся по рёбрам)

Задан граф. Необходимо найти наибольшее число не пересекающихся по рёбрам маршрутов между заданными вершинами  $v$  и  $w$  графа.

### Формат входных данных

Первая строка содержит число  $N$  вершин и число  $M$  рёбер графа ( $1 \leq N \leq 100$ ,  $0 \leq M \leq 4950$ ). Вершины графа пронумерованы целыми числами от 1 до  $N$ . Каждая из следующих  $N$  строк файла задаёт множество вершин графа, смежных с очередной вершиной. Так,  $(i + 1)$ -я строка файла содержит вершины, смежные с вершиной  $i$ , последним в строке идёт число 0 (если нет вершин, смежных с некоторой вершиной графа, то соответствующая строка содержит только число 0). В последней строке файла находятся вершины  $v$  и  $w$ . Все числа внутри одной строки разделены одним или несколькими пробелами.

### Формат выходных данных

В единственной строке выведите наибольшее число не пересекающихся по рёбрам маршрутов между вершинами  $v$  и  $w$ .

#### Пример

входной файл	выходной файл	пояснение
<pre> 6 9 2 3 6 0 1 3 0 1 2 4 6 0 3 5 6 0 4 6 0 1 3 4 5 0 1 4                     </pre>	3	рис. 1.24

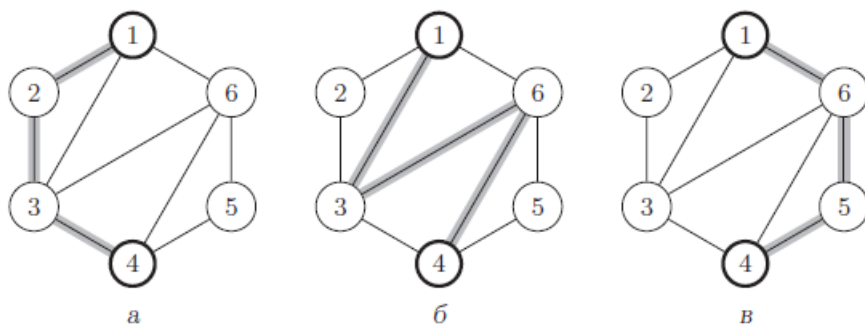


Рис. 1.24

## Задача 2. Лабиринт (без пересечений по дорогам)

Лабиринт задаётся матрицей смежности  $C$  размера  $N \times N$ , где  $C_{i,j} = 1$ , если комната  $i$  связана с комнатой  $j$  посредством дороги ( $i$  и  $j$  — целые числа от 1 до  $N$ ). Часть комнат является входами, часть — выходами. Входы и выходы задаются наборами комнат  $\{X_1, \dots, X_p\}$  и  $\{Y_1, \dots, Y_k\}$  соответственно. Необходимо найти наибольшее число людей, которых можно провести от входов до выходов таким образом, чтобы их пути не пересекались по дорогам, но могли пересекаться по комнатам.

### Формат входных данных

Первая строка содержит размер лабиринта  $N$  ( $1 \leq N \leq 300$ ), число входов  $p$  и число выходов  $k$ . Ни одна из комнат не является одновременно и входом, и выходом. Затем идут  $N$  строк, которые содержат матрицу смежности  $C$  (каждой строке матрицы соответствует строка входного файла). Следующая строка содержит  $p$  различных чисел — номера входов. Последняя строка содержит  $k$  различных чисел — номера выходов. Комнаты нумеруются числами от 1 до  $N$ .

#### Формат выходных данных

Единственная строка этого файла содержит наибольшее число людей, которых можно провести.

#### Пример

входной файл	выходной файл	пояснение
<pre> 6 1 3 0 1 1 0 0 1 1 0 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 4 5 6 </pre>	3	рис. 1.25

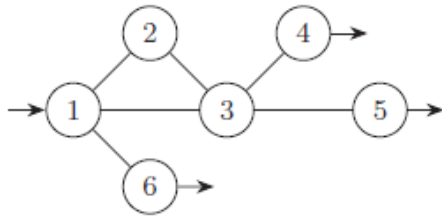


Рис. 1.25

### Задача 3. Шестерёнки

Каждой из  $N$  шестерёнок присвоен номер: число от 1 до  $N$ . Задано  $M$  соединений пар шестерёнок в виде  $\{i, j\}$  (шестерёнка с номером  $i$  находится в зацеплении с шестерёнкой с номером  $j$ ).

Необходимо определить, какое наименьшее число шестерёнок нужно удалить, чтобы первая пришла в движение. Если таких вариантов несколько, то нужно выбрать тот, при котором вращаться будет наибольшее число шестерёнок.

#### Формат входных данных

Первая строка содержит два числа, записанных через пробел: число  $N$  шестерёнок и число  $M$  соединений пар ( $0 \leq N \leq 10, 0 \leq M \leq 45$ ). Затем идут  $M$  строк файла, каждая из которых задаёт пару  $\{i, j\}$  соединённых шестерёнок ( $1 \leq i < j \leq N$ ).

#### Формат выходных данных

Если первая шестерёнка пришла в движение без удаления других шестерёнок, то в первой строке выведите Yes, а через пробел укажите число шестерёнок, которые пришли при этом в движение; во второй строке укажите (через пробел) номера шестерёнок, которые пришли в движение (в порядке возрастания их номеров). Если для вращения первой шестерёнки необходимо удаление других шестерёнок, то в первой строке выведите No, а во второй — число шестерёнок, которые нужно убрать, и через двоеточие — их номера (если такой набор не один, то выдаётся лексикографически наименьший; в наборе шестерёнки перечисляются в порядке возрастания их номеров); в третьей строке — число шестерёнок, которые пришли в движение, и через двоеточие — их номера (в порядке возрастания).

**Пример**

входной файл	выходной файл
6 10 1 2 2 3 3 4 4 5 5 6 6 1 1 4 2 5 3 6 2 6	No 1:2 5:1 3 4 5 6

**Задача 4. Открытки и конверты**

Имеется  $N$  прямоугольных конвертов и  $N$  прямоугольных открыток различных размеров. Необходимо определить, можно ли разложить все открытки по конвертам, чтобы в каждом конверте было по одной открытке. Открытки нельзя складывать, сгибать и т. п., но можно помещать в конверт под углом. Например, открытка с размерами сторон  $5 \times 1$  помещается в конверты с размерами  $5 \times 1$ ,  $6 \times 3$ ,  $4,3 \times 4,3$ , но не входит в конверты с размерами  $4 \times 1$ ,  $10 \times 0,5$ ,  $4,2 \times 4,2$ .

**Формат входных данных**

Первая строка содержит число  $N$  — число открыток (конвертов) ( $1 \leq N \leq 500$ ). Затем идут  $N$  строк, каждая из которых содержит два действительных числа, задающих размеры очередной открытки. Затем следуют  $N$  строк файла, каждая из которых содержит два действительных числа, задающих размеры очередного конверта. Все числа положительны, не превосходят 1000 и записаны с не более чем одним знаком после точки.

**Формат выходных данных**

Если все открытки можно разложить, то единственная строка файла должна содержать сообщение YES. Если все открытки нельзя разложить, то первая строка выходного файла должна содержать сообщение NO, а вторая — наибольшее число открыток, которые можно разложить по конвертам.

**Пример**

входной файл	выходной файл
2 7 1.4 6.5 5 7 7 6 6	YES
1 5 20 16 19	YES
3 7 1.4 6.5 5 4 4 7 7 6 6 5 3	NO 2

**Задача 5. Разбиение на команды по численности**

Необходимо определить, можно ли разбить  $N$  студентов на две команды, численность которых отличается не более чем в два раза, если известно, что в каждой команде любых два студента должны быть знакомы друг с другом. Круг знакомств задаётся матрицей  $A$  размера  $N \times N$  с элементом  $A_{ij}$ , равным 1, если  $i$ -й студент знаком со студентом  $j$ , и элементом  $A_{ij}$ , равным 0, если  $i$ -й студент не знаком со студентом  $j$ .

**Формат входных данных**

Первая строка содержит число  $N$  студентов ( $1 \leq N \leq 1000$ ). Затем идут  $N$  строк, которые задают матрицу  $A$  знакомств (каждой строке матрицы соответствует отдельная строка

входного файла, числа в строках разделены пробелами).

#### Формат выходных данных

Если можно разбить студентов на две команды, численность которых отличается не более чем в два раза, то в первой строке выведите YES, а во второй — номера (в порядке возрастания) студентов, которые попали в команду с наибольшей численностью.

Если нельзя разбить студентов на две команды, численность которых отличается не более чем в два раза, то единственная строка файла должна содержать сообщение NO.

#### Примеры

входной файл	выходной файл
<pre>7 0 1 1 1 1 0 0 1 0 1 1 1 0 0 1 1 0 1 1 0 0 1 1 1 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0</pre>	NO
<pre>6 0 1 1 1 0 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0</pre>	<pre>YES 1 2 3 4</pre>
<pre>2 0 0 0 0</pre>	<pre>YES 2</pre>

#### Задача 6. Разбиение на группы незнакомых

Имеется  $N$  человек и матрица  $A$  размера  $N \times N$ . Элемент  $A_{ij}$  матрицы равен 1, если человек  $i$  знаком с человеком  $j$  (если  $i$ -й человек знает  $j$ -го, то считаем, что и  $j$ -й человек знает  $i$ -го), и элемент  $A_{ij}$  матрицы равен 0, если  $i$ -й человек не знаком с человеком  $j$ . Можно ли разбить людей на две группы, чтобы в каждой группе были только незнакомые люди (в каждой группе должен быть хотя бы один человек)?

#### Формат входных данных

Первая строка содержит число  $N$  людей ( $2 \leq N \leq 500$ ). Затем идут  $N$  строк файла, которые задают матрицу  $A$  знакомств (каждой строке матрицы соответствует отдельная строка входного файла, числа в строках разделены пробелами).

#### Формат выходных данных

Если можно разбить людей на две группы, чтобы в каждой группе были только незнакомые люди, то выведите в первой строке YES, а во второй — номера людей (люди нумеруются от 1 до  $N$ ), которые попали в одну из групп. Числа в строке упорядочены по возрастанию. Если разбить нужным образом нельзя, то выведите NO.

#### Примеры

входной файл	выходной файл
<pre>4 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 0</pre>	NO
<pre>4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</pre>	<pre>YES 1 2</pre>

#### Задача 7. Олимпиада

На олимпиаду прибыли  $N$  человек. Некоторые из них знакомы между собой. Круг знакомств задан матрицей  $A$  размера  $N \times N$ . Элемент  $A_{ij}$  матрицы равен 1, если человек  $i$  знаком

с человеком  $j$  (если  $i$ -й человек знает  $j$ -го, то это значит, что  $j$ -й человек знает  $i$ -го), и элемент  $A_{ij}$  матрицы равен 0, если  $i$ -й человек не знаком с человеком  $j$ . Необходимо определить, можно ли опосредованно (незнакомые люди могут познакомиться только через общего знакомого) познакомиться всех между собой.

#### Формат входных данных

Первая строка содержит число  $N$  людей ( $2 \leq N \leq 500$ ). Затем идут  $N$  строк файла, которые задают матрицу знакомств  $A$  (каждой строке матрицы соответствует отдельная строка входного файла).

#### Формат выходных данных

Если можно опосредованно познакомиться всех прибывших на олимпиаду между собой, то первая строка файла будет содержать сообщение YES; в противном случае строка файла должна содержать сообщение NO.

#### Пример

входной файл	выходной файл
<pre> 5 0 1 1 0 0 1 0 1 0 1 1 1 0 1 0 0 0 1 0 0 0 1 0 0 0 </pre>	YES
<pre> 4 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 </pre>	NO
<pre> 2 0 1 1 0 </pre>	YES

### Задача 8. Станки

Конвейер состоит из  $N$  различных станков. Есть  $N$  рабочих. Известна матрица  $C$  размера  $N \times N$ , где элемент  $c_{ij}$  задаёт производительность  $i$ -го рабочего на  $j$ -м станке. Необходимо определить, каким должно быть распределение рабочих по станкам (каждый рабочий может быть назначен только на один станок, на каждом станке может работать только один рабочий), чтобы производительность всего конвейера была максимальной. Производительность конвейера при некотором распределении рабочих по станкам равна минимальной производительности рабочих на назначенных им на конвейере станках. Если решение не единственно, вывести решение, первое в лексикографическом порядке среди всех решений.

#### Формат входных данных

Первая строка содержит число рабочих (станков)  $N$  ( $1 \leq N \leq 500$ ). Затем идут  $N$  строк файла, которые задают матрицу  $C$  производительностей ( $1 \leq c_{ij} \leq 1000$ ).

#### Формат выходных данных

В первой строке выведите максимальную возможную производительность конвейера. Во второй строке — номера станков, на которые должны быть распределены рабочие  $1, 2, \dots, N$  соответственно.

#### Пример

входной файл	выходной файл
<pre> 4 4 2 6 1 5 2 3 3 4 7 1 4 4 3 2 5 </pre>	<pre> 5 3 1 2 4 </pre>

### Задача 9. Встреча

В городе  $N$  домиков и  $M$  дорог. В каждом домике живёт по одному человеку. Необходимо найти точку (место встречи всех людей), от которой суммарное расстояние по дорогам до всех домиков будет минимальным. Место встречи следует искать среди точек, в которых распо-

ложены домики, а также точек, лежащих на дороге и отстоящих от домиков на целое число единиц длины.

#### Формат входных данных

Первая строка содержит число  $N$  домиков ( $1 \leq N \leq 100$ ) и число  $M$  дорог ( $0 \leq M \leq N \times (N - 1)/2$ ). Домики пронумерованы числами от 1 до  $N$ . Затем идут  $M$  строк файла, по три целых числа в каждой, задающие дороги: номера  $u$  и  $v$  домиков ( $1 \leq u, v \leq N$ ), которые являются концами дороги, и длина  $w$  дороги ( $1 \leq w \leq 100000$ ). Гарантируется, что никакая дорога не соединяет домик с самим собой. Между любой парой домиков имеется не более одной дороги. По дорогам можно двигаться в обе стороны.

#### Формат выходных данных

Если точка встречи лежит на дороге, то выведите три целых числа: номера конечных домиков дороги и расстояние от первого из них до этой точки. Если точка совпадает с домом, то выведите его номер и суммарное расстояние от всех домиков до него. Гарантируется, что решение существует. Если решений несколько, выведите любое.

#### Пример

входной файл	выходной файл
<pre> 6 9 1 2 2 2 3 3 3 4 15 4 5 5 5 6 6 6 1 20 1 3 7 3 6 5 4 6 8 2 1 1 2 10 </pre>	<pre> 3 37 1 2 5 </pre>

#### Замечание

В первом примере ответ «6 37» также будет считаться верным.

#### Задача 10. Янка

Янка положил на стол  $N$  выпуклых  $K$ -гранников и  $N$  различных типов наклеек по  $K$  штук каждая. Ночью кто-то наклеил наклейки на грани, по одной на грань (на одном и том же многограннике могло оказаться несколько наклеек одного типа). Янке необходимо расставить многогранники так, чтобы наклейка каждого типа была видна ровно  $K - 1$  раз (наклейка не видна, если она расположена на грани, на которой стоит многогранник). Предполагается, что такая расстановка всегда существует.

#### Формат входных данных

Первая строка содержит два целых числа:  $N$  ( $1 < N \leq 500$ ) и  $K$  ( $1 < K \leq 100$ ). Затем идут  $N$  строк файла по  $K$  чисел в каждой. Каждая строка соответствует одному  $K$ -граннику и содержит номера наклеек, которые наклеены на 1-ю, 2-ю, ...,  $K$ -ю грани данного многогранника соответственно.

#### Формат выходных данных

Выведите в одну строку искомую расстановку:  $i$ -й элемент этой строки — номер наклейки на  $i$ -м многограннике, на которую его поставили (в результате в этой строке каждый тип из  $N$  типов наклеек встретится ровно один раз). Числа в строке разделены одним пробелом.

#### Пример

входной файл	выходной файл	пояснение
<pre> 4 4 1 2 1 1 2 3 3 3 3 2 1 2 4 4 4 4 </pre>	<pre> 2 3 1 4 </pre>	рис. 1.26

*K*-границы    Типы наклеек

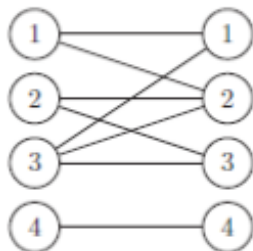


Рис. 1.26

### Задача 11. Второй по длине маршрут

Задано  $N$  городов с номерами от 1 до  $N$  и сеть из  $M$  дорог с односторонним движением между ними. Каждая дорога определяется тройкой  $(i, j, k)$  натуральных чисел, где  $i$  — номер города, в котором дорога начинается,  $j$  — номер города, в котором дорога заканчивается, а  $k$  — её длина. Дороги друг с другом могут пересекаться только в конечных городах. Все ориентированные маршруты между двумя указанными городами  $A$  и  $B$  можно упорядочить в список по неубыванию их длин. Необходимо найти один из маршрутов, который является вторым в этом упорядоченном списке (между городами  $A$  и  $B$  существует по крайней мере два ориентированных маршрута). Вывести его длину  $L$  и города, через которые он проходит (искомый маршрут по дугам может проходить несколько раз).

#### Формат входных данных

Первая строка содержит два целых числа  $N$  и  $M$  ( $1 \leq N \leq 10000$ ,  $1 \leq M \leq 100000$ ). Затем идут  $M$  строк файла по три числа в каждой:  $i, j, k$ . Последняя строка содержит номера городов  $A$  и  $B$ . Формат выходных данных Первая строка должна содержать длину второго по длине маршрута между городами  $A$  и  $B$ . Вторая строка — это номера городов, через которые проходит второй по длине маршрут. В случае если вторых по длине маршрутов несколько, выдать в выходной файл информацию о любом из них.

#### Пример

входной файл	выходной файл
<pre> 3 3 1 2 1 2 1 1 2 3 1 1 3 </pre>	<pre> 4 1 2 1 2 3 </pre>
<pre> 3 4 1 3 10 2 1 10 2 3 20 3 2 5 2 3 </pre>	<pre> 20 2 3 </pre>

### Задача 12. Чётно-нечётная магистраль

Имеется страна с  $N$  городами. Между некоторыми парами городов построены магистрали. Длина любого прямого соединения равна единице. Движение по магистрали возможно в обе стороны. Систему магистралей будем называть чётно-нечётной, если каждая пара различных городов соединена маршрутом как чётной длины, так и нечётной (причём маршрут может проходить через один город несколько раз). Необходимо определить, является ли система магистралей чётно-нечётной. Если ответ на вопрос отрицательный, то нужно найти одно из подмножеств множества городов, в котором наибольшее число элементов и которое удовлетворяет следующему условию: если какие-либо два различных города из этого подмножества соединены маршрутом, то его длина чётна.

#### Формат входных данных

Первая строка содержит число  $N$  городов ( $1 \leq N \leq 300$ ). Следующие  $N$  строк файла задают магистрали:  $(i + 1)$ -я строка содержит номера городов, которые связаны напрямую

с городом  $i$  (если таких городов нет, то строка содержит нуль; числа в строке разделены пробелами).

#### Формат выходных данных

Если система магистралей является чётно-нечётной, то выведите единственную строку YES. В противном случае в первой строке выведите NO, а во второй — мощность описанного наибольшего подмножества.

#### Примеры

входной файл	выходной файл
5 2 5 3 1 2 4 3 5 4 1	YES
3 2 1 0	NO 2
5 2 3 1 3 1 2 5 4	NO 2

#### Задача 13. Пересекающиеся дороги

Сеть дорог определяется следующим образом. Имеется  $N$  перекрёстков и  $K$  дорог, связывающих перекрёстки. Каждая дорога определяется тройкой чисел: двумя номерами перекрёстков и временем, требующимся на проезд по этой дороге. Необходимо найти кратчайший маршрут машины от перекрёстка  $I$  до перекрёстка  $J$ , если на перекрёстке машина должна выждать время, равное числу пересекающихся дорог (для начального и конечного перекрёстков маршрута ожидание не требуется). Движение по дороге возможно в обоих направлениях.

#### Формат входных данных

В первой строке находятся два числа:  $N$  ( $1 \leq N \leq 10000$ ) и  $K$  ( $1 \leq K \leq 100000$ ). Во второй строке заданы номера  $I$  и  $J$  перекрёстков, между которыми надо найти кратчайший маршрут. Каждая из следующих  $K$  строк содержит по три числа (через пробел): номера перекрёстков, задающих очередную дорогу, и время, требующееся на проезд по этой дороге (это число является неотрицательным и не превосходит 100 000). Между любыми двумя перекрёстками — не более одной дороги.

#### Формат выходных данных

Выведите в первой строке минимальное время, а во второй — номера перекрёстков при движении от перекрёстка  $I$  до перекрёстка  $J$ , соответствующие найденному кратчайшему маршруту. Гарантируется, что маршрут между заданными перекрёстками существует.

#### Пример

входной файл	выходной файл
7 11 1 6 1 2 1 2 3 1 3 4 1 4 5 1 5 6 1 1 7 6 2 7 6 3 7 6 4 7 6 5 7 6 6 7 6	17 1 2 3 4 5 6

#### Задача 14. Домики



На прямоугольном участке расположены  $N$  домиков. Левый нижний угол участка имеет координаты  $(0, 0)$ , а правый верхний —  $(100, 100)$ . Местоположение каждого домика задаётся целочисленными координатами его нижнего левого угла. Каждый домик имеет размер  $5 \times 5$  м. Стороны домиков параллельны сторонам участка. Домики отстоят друг от друга не менее чем на 1 м. Необходимо найти один из кратчайших путей от точки  $(0, 0)$  до точки  $(100, 100)$ . При движении можно затронуть только стены домиков. Найденный путь представить в виде координат концов прямолинейных отрезков, составляющих этот путь.

#### Формат входных данных

Первая строка содержит число  $N$  домиков ( $0 \leq N \leq 30$ ). Следующие  $N$  строк содержат координаты домиков:  $(i + 1)$ -я строка задаёт координаты левого нижнего угла  $i$ -го домика.

#### Формат выходных данных

В первой строке необходимо вывести длину кратчайшего пути (с абсолютной погрешностью не более  $0,1$ ), а во второй — координаты концов прямолинейных отрезков этого пути (см. пример).

#### Пример

входной файл	выходной файл	пояснение
6 5 5 15 13 15 20 30 20 35 40 50 50	142.2 (0;0) (10;5) (20;13) (30;25) (55;50) (100;100)	рис. 1.27

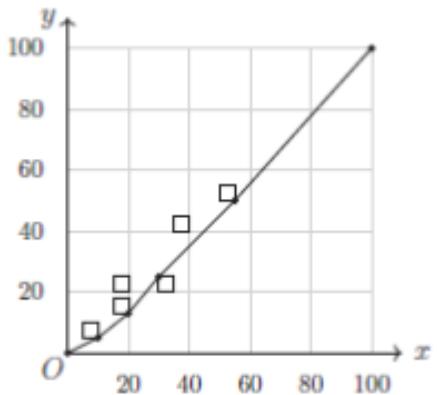


Рис. 1.27

#### Задача 14. Хакеры

Винни-Пух и Пятачок решили защитить компьютерную сеть от хакеров, которые выкачивали оттуда секретную информацию. Компьютерная сеть Винни-Пуха и Пятачка состояла из соединённых между собой больших ЭВМ, к каждой из которых подключалось несколько терминалов. Подключение к одной из больших ЭВМ позволяет получить информацию, содержащуюся в памяти этой ЭВМ, а также всю доступную информацию, которую может запросить данная ЭВМ. Хакеры и раньше нападали на подобные компьютерные сети, и их тактика была известна. Поэтому Винни-Пух и Пятачок разработали специальную программу, которая помогла принять меры против нападения. Тактика хакеров: при нападении они всегда получают доступ к информации всех ЭВМ в сети. Добиваются они этого, захватывая некоторые ЭВМ в сети, чтобы от них можно было запросить информацию, которая имеется у оставшихся ЭВМ. Вариантов захвата существует множество, например захватить все ЭВМ. Однако хакеры всегда выбирают такой вариант, при котором суммарное число терминалов у захваченных ЭВМ минимально. Необходимо определить список номеров ЭВМ, которые могут быть выбраны хакерами для захвата сети согласно их тактике.

#### Формат входных данных

Первая строка содержит число  $N$  ЭВМ в сети ( $1 \leq N \leq 100000$ ). Каждая из следующих  $N$  строк содержит число  $T_i$  терминалов у машины с номером  $i$  ( $1 \leq T_i \leq 1000$ ). Далее идут  $K$  строк (список прав на запрос) по два числа в каждой:  $A_j$  и  $B_j$  — номера машин, соединённых между собой. Каждая пара  $(A_j, B_j)$  означает, что ЭВМ с номером  $A_j$  имеет право запрашивать информацию, которая имеется у ЭВМ с номером  $B_j$  ( $A_j = B_j$ ). Пары не повторяются. Завершает файл строка, содержащая два нуля.

#### Формат выходных данных

В первой строке выведите число захватываемых машин, во второй — их номера в произвольном порядке. Если решений несколько, выведите любое из них.

#### Пример

входной файл	выходной файл	пояснение
<pre> 7 8 7 8 3 6 1 9 1 2 2 3 3 4 4 5 5 1 2 5 3 4 7 6 0 0 </pre>	<pre> 2 4 7 </pre>	рис. 1.28

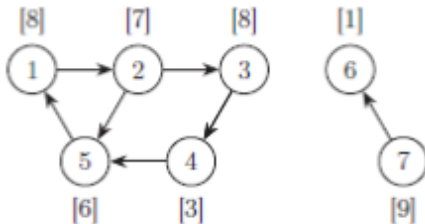


Рис. 1.28

#### Задача 15. Пирамида Хеопса

Внутри пирамиды Хеопса имеется  $N$  комнат, в которых установлены  $2M$  модулей, составляющих  $M$  устройств. Каждое устройство включает два модуля, располагающихся в разных комнатах, и предназначено для перемещения между парой комнат, в которых установлены эти модули. Перемещение происходит за 0,5 условных единиц времени. В начальный момент времени модули всех устройств переходят в подготовительный режим. Каждый из модулей имеет некоторый свой целочисленный период времени, в течение которого он находится в подготовительном режиме. По истечении этого времени модуль мгновенно включается, после чего опять переходит в подготовительный режим. Устройством можно воспользоваться только в тот момент, когда одновременно включаются оба его модуля. Индиана Джонс сумел проникнуть в гробницу фараона. Обследовав её, он включил устройства и собрался уходить, но в этот момент проснулся хранитель гробницы. Теперь Джонсу необходимо как можно быстрее попасть в комнату с номером  $N$ , в которой находится выход из пирамиды. При этом из комнаты в комнату он может попадать только при помощи устройств, так как проснувшийся хранитель закрыл все двери в комнатах пирамиды. Необходимо написать программу, которая получает на входе описание расположения устройств, их характеристик и выдаёт значение оптимального времени и последовательность устройств, которыми надо воспользоваться, чтобы попасть из комнаты 1 в комнату  $N$  за это время. Формат входных данных В первой строке содержится число  $N$  комнат ( $2 \leq N \leq 10000$ ). Во второй строке — число  $M$  устройств ( $0 \leq M \leq 100000$ ).

В следующих  $M$  строках находится информация об устройствах. Каждая строка содержит числа  $R_{1,i}$ ,  $T_{1,i}$ ,  $R_{2,i}$ ,  $T_{2,i}$  — информацию об устройстве  $i$ : •  $R_{1,i}$  и  $R_{2,i}$  — номера комнат, в которых располагаются модули устройства  $i$ ; •  $T_{1,i}$ ,  $T_{2,i}$  — периоды времени, через которые включаются соответствующие модули ( $R_{1,i}$  — в момент времени  $T_{1,i}$  и  $R_{2,i}$  — в момент времени  $T_{2,i}$ ) ( $1 \leq T_{1,i}, T_{2,i} \leq 100000$ , целые числа).

#### Формат выходных данных

Первая строка выходного файла содержит оптимальное время (с абсолютной погрешностью не более  $0,1$ ). Во второй строке указана последовательность устройств, которыми надо воспользоваться, чтобы попасть из комнаты 1 в комнату  $N$  за минимальное время. Если решений несколько, выведите любое. Гарантируется, что решение существует.

#### Пример

входной файл	выходной файл
5 5 1 6 2 4 2 1 3 7 3 1 4 1 4 2 5 8 2 2 4 9	16.5 1 2 3 4

### Задача 16. Кратчайший маршрут с дополнительным временем преодоления перекрёстка

План города представляет собой множество перекрёстков, соединённых дорогами (по дороге движение разрешено в обоих направлениях). Каждая дорога задаётся номерами перекрёстков, которые она соединяет, и временем движения по ней. Между двумя различными перекрёстками может быть не более одной дороги. Дорога не может соединять перекрёсток сам с собой. Кроме того, время преодоления перекрёстка  $I$  равно  $qk_i$ , где  $q$  — заданная константа, а  $k_i$  — число дорог, подходящих к перекрёстку  $i$ . Необходимо найти кратчайший по времени маршрут от перекрёстка  $s$  до перекрёстка  $f$ . Конечный перекрёсток  $f$  преодолевать не надо, а стартовая вершина  $s$  маршрута преодолевается как перекрёсток.

#### Формат входных данных

В первой строке находятся числа  $N$  и  $M$  — число перекрёстков ( $1 \leq N \leq 11000$ ) и число дорог на плане города ( $0 \leq M \leq 100000$ ) соответственно. В каждой из следующих  $M$  строк файла сначала расположены номера перекрёстков, которые связывает очередная дорога, а затем время движения по ней (от 0 до 1000). В последней строке находятся номера перекрёстков  $s$  и  $f$  ( $1 \leq s, f \leq N$ ), а также константа  $q$  ( $1 \leq q \leq 100$ ).

#### Формат выходных данных

Если проехать от перекрёстка  $s$  до перекрёстка  $f$  можно, в первой строке выведите сообщение Yes, а во второй строке — минимальное время движения. Если проехать нельзя, то в единственной строке выведите сообщение No.

#### Пример

входной файл	выходной файл
6 9 1 2 2 1 3 1 2 4 1 2 5 1 3 4 3 3 5 1 4 5 2 4 6 1 5 6 1 1 6 1	Yes 12

### Задача 17. Железные и шоссейные дороги

Имеется  $N$  городов, пронумерованных числами от 1 до  $N$ . Некоторые из них соединены двусторонними дорогами, пересекающимися только в городах. Имеется два типа дорог: шоссей-

ные и железные. Для каждой дороги известна базовая стоимость проезда по ней. Необходимо проехать из города А в город В, уплатив минимальную сумму за проезд. Стоимость проезда зависит от набора проезжаемых дороги от способа проезда. Так, если вы подъехали к городу С по шоссе (железной) дороге  $X \rightarrow C$  и хотите ехать дальше по дороге  $C \rightarrow Y$  того же типа, то должны уплатить только базовую стоимость проезда по дороге  $C \rightarrow Y$ . Если тип дороги  $C \rightarrow Y$  отличен от типа дороги  $X \rightarrow C$ , то нужно уплатить базовую стоимость проезда по дороге  $C \rightarrow Y$  плюс 10% от базовой стоимости проезда по этой дороге (страховой взнос). При выезде из города А страховой взнос платится всегда. Необходимо найти стоимость самого дешёвого маршрута, если он существует.

#### Формат входных данных

В первой строке находится целое число  $N$  городов ( $1 \leq N \leq 1000$ ). Во второй строке задано целое число  $M$  дорог ( $0 \leq M \leq 1000000$ ). В каждой из следующих  $M$  строк находятся четыре числа  $x, y, t, p$  (разделённые пробелом), где  $x$  и  $y$  — номера городов, которые соединяет дорога,  $t$  — тип дороги (0 — шоссе, 1 — железная),  $p$  — базовая стоимость проезда по ней ( $p$  — действительное число,  $0 < p \leq 1000$ ). В последней строке задаются номера А и В начального и конечного городов.

#### Формат выходных данных

Если маршрута не существует, то в единственной строке выведите No. Если же маршрут существует, то в первой строке выведите Yes, а во второй — стоимость проезда по самому дешёвому маршруту (с точностью до двух знаков после точки).

#### Пример

входной файл	выходной файл
<pre> 5 5 1 5 1 10 1 3 1 10 1 4 0 30 1 2 0 1000 4 3 0 10 5 2 </pre>	<pre> Yes 1062.00 </pre>

### Задача 18. Кратчайший путь в зависимости от направления поворота

Заданы декартовы координаты  $N$  перекрёстков города, которые пронумерованы числами от 1 до  $N$ . На каждом перекрёстке имеется светофор. Некоторые перекрёстки соединены дорогами с двусторонним (правосторонним) движением, пересекающимися только на перекрёстках. Для каждой из  $M$  дорог известно неотрицательное время, требуемое для проезда по ней от одного перекрёстка до другого. Необходимо проехать от перекрёстка А до перекрёстка В за минимальное время. Время проезда зависит от набора проезжаемых дороги от времени ожидания на перекрёстках. Так, если вы подъехали от перекрёстка  $X$  к перекрёстку  $C$  по дороге  $X \rightarrow C$  и хотите ехать дальше по дороге  $C \rightarrow Y$ , то время ожидания на перекрёстке  $C$  зависит от того, поворачиваете вы налево или нет. Если вы поворачиваете налево, то время ожидания равно  $DK$ , где число  $D$  — количество дорог, пересекающихся на перекрёстке  $C$ , а число  $K$  — некоторая константа. Если вы не поворачиваете налево, то время ожидания на перекрёстке равно нулю. Разворот на перекрёстке считается поворотом налево. Замечание Если первоначально машина находится на перекрёстке с координатами  $(x, y)$ , то предполагается, что она приехала в эту точку из точки с координатами  $(x, y - 1)$ .

#### Формат входных данных

В первой строке входного файла находятся натуральные числа  $N, M$  и  $K$  ( $N \leq 1000, M \leq 1000, K \leq 1000$ ). В каждой из следующих  $N$  строк — пара целых чисел  $x$  и  $y$  (разделённых пробелом), которые задают координаты перекрёстка города ( $|x|, |y| < 1000$ ). В каждой из  $M$  следующих строк — по три числа  $p, r$  и  $t$  (разделённые пробелами), где  $p$  и  $r$  — номера перекрёстков, которые соединяет дорога, а натуральное число  $t$  ( $t \leq 1000$ ) — время проезда по ней. В последней строке файла находятся номера А и В начального и конечного перекрёстков

соответственно.

#### Формат выходных данных

Если пути не существует, то единственная строка выходного файла должна содержать сообщение No. Если путь существует, то в первой строке нужно вывести сообщение Yes, а во второй — натуральное число  $t$  — время проезда по самому быстрому маршруту.

#### Пример

входной файл	выходной файл
<pre> 6 5 5 4 2 4 0 2 4 4 6 6 4 0 4 1 2 0 3 4 1 1 5 1 1 3 1 6 3 1 2 6 </pre>	<pre> Yes 16 </pre>

#### Задача 19. Прогулка

Хозяин вышел на прогулку с собакой. Известно, что путь хозяина представляет собой ломаную линию, координаты отрезков ломаной заданы:  $(X_i, Y_i)$ ,  $i = 1, \dots, N$ . Точка  $(X_1, Y_1)$  — начальная точка ломаной линии, а точка  $(X_N, Y_N)$  — конечная точка ломаной. Хозяин во время прогулки движется от стартовой точки по отрезкам ломаной линии и заканчивает путь в конечной точке ломаной. У собаки есть свои любимые места (координаты любимых мест заданы:  $(X'_j, Y'_j)$ ,  $j = 1, \dots, M$ ), которые собака хотела бы посетить. В то время как хозяин проходит один отрезок ломаной, собака может посетить только одно из своих любимых мест. В начальной и конечной точке ломаной, а также когда хозяин проходит каждую точку изгиба ломаной, собака обязана подбежать к хозяину. Известно, что скорость собаки в два раза выше скорости хозяина. Необходимо определить, какое наибольшее число своих любимых мест сможет посетить собака за время прогулки.

#### Формат входных данных

Первая строка содержит два числа:  $N$  — число точек ломаной (включая начальную и конечную точки пути) и  $M$  — число любимых мест собаки ( $1 \leq N \leq 100$ ,  $0 \leq M \leq 100$ ). В следующих  $N$  строках идут координаты точек ломаной: сначала координата  $x$ , а затем через пробел — координата  $y$  и т.д. (координаты точек ломаной следуют в той последовательности, в которой они соединяются отрезками, начиная от начальной точки и заканчивая конечной). В следующих  $M$  строках — координаты любимых мест собаки: сначала координата  $x$ , а затем через пробел — координата  $y$  и т.д. Координаты — целые числа, по модулю не превосходящие 10 000. Все точки попарно различны.

#### Формат выходных данных

Выведите в единственной строке два числа (через пробел). Первое — наибольшее количество мест, которые сможет посетить собака за время прогулки (это и все точки ломаной, и некоторые любимые места собаки), второе — наибольшее число любимых мест собаки, которые она смогла посетить.

### Пример

входной файл	выходной файл	пояснение
6 5 1 2 4 4 9 1 16 4 20 2 22 2 2 4 6 2 8 3 12 3 12 1	9 3	рис. 1.30



Рис. 1.30

### Задача 20. Сборка прибора

Прибор спроектирован таким образом, что он будет собираться из отдельных узлов, причём каждый узел уникален. Сами узлы, в свою очередь, могут требовать предварительной сборки. Для сборки каждого узла необходимо, чтобы все узлы, комплектующие его, уже были собраны. Узлы, не требующие сборки, обязательно тестируются на работоспособность. Собираемые узлы тестировать не требуется. На сборку одного узла или на его тестирование тратится один день. Готовый узел должен быть помещён на склад и может быть взят со склада только тогда, когда он необходим для сборки очередного узла или самого прибора. Хранение узла на складе в течение одного дня требует оплаты в размере одной условной денежной единицы. Необходимо организовать сборку таким образом, чтобы плата за аренду склада была минимальной.

#### Формат входных данных

Первая строка содержит общее число  $N$  узлов (узлы пронумерованы от 1 до  $N$ ,  $N \leq 50$ ). Вторая строка — номер узла, который надо собрать. Каждая из следующих  $N$  строк файла содержит информацию об отдельном узле:  $L$  — его номер,  $C$  — количество его комплектующих. Затем в этой же строке идут  $C$  чисел, которые задают номера комплектующих узлов для узла  $L$ . Разделителем чисел в пределах строки является символ «:» (двоеточие).

#### Формат выходных данных

Выведите в первой строке плату за аренду, а во второй — последовательность, в которой собираются узлы (если вариантов сборки несколько, нужно вывести ту, которая лексикографически меньше).

### Пример

входной файл	выходной файл
6 5 1:1:6 2:0 3:2:1:4 4:0 5:2:2:3 6:0	7 6 1 4 3 2 5

### Задача 21. Скрудж Мак-Дак

Скрудж Мак-Дак решил сделать прибор для управления самолётом. Как известно, положение штурвала зависит от состояния входных датчиков, но эта функция довольно сложна. Его механик делает устройство, вычисляющее эту функцию за несколько этапов с использованием промежуточной памяти и вспомогательных функций. Для вычисления каждой из функций требуется, чтобы в ячейках памяти уже находились вычисленные параметры (которые являются значениями вычисленных функций), необходимые для её вычисления. Вы-

числение функции без параметров может производиться в любое время. После вычисления функции ячейки могут быть использованы повторно (хотя бы для записи результата вычисленной функции). Структура вызова функций такова, что каждая функция вычисляется не более одного раза и любой параметр используется не более одного раза. Любой параметр есть имя функции. Поскольку Скрудж не хочет тратить лишних денег на микросхемы, он поставил задачу минимизировать память прибора. По заданной структуре вызовов функций нужно определить минимально возможный размер памяти прибора и указать последовательность вычисления функций. Первоначально необходимо вычислить функции, которые принимают на вход наибольшее число параметров. При неоднозначности выбора функции требуется вычислять в лексикографическом порядке.

#### Формат входных данных

Первая строка содержит общее число  $N$  функций ( $1 \leq N \leq 10000$ ). Вторая строка — имя функции, которую необходимо вычислить (имя функции есть натуральное число от 1 до  $N$ ). Каждая из следующих  $N$  строк содержит имя функции, число параметров и список имён параметров.

#### Формат выходных данных

В первой строке выведите минимальное число ячеек памяти, во второй — последовательность вычисления функций.

#### Пример

входной файл	выходной файл
5	1
2	3 1 5 4 2
2:1:4	
1:1:3	
5:1:1	
3:0	
4:1:5	

### Задача 22. Заправочные станции

Имеется  $N$  городов, соединённых  $M$  двусторонними дорогами. Для каждой дороги задана её протяжённость в километрах. Машина может поворачивать (изменять направление движения) только в городах. Машина имеет бак вместимостью  $Z$  литров бензина, и для неё задан расход бензина  $X$  литров на один километр. В некоторых городах имеются заправочные станции. У каждой заправочной станции задана своя стоимость дозаправки  $S_i$  до полного бака (т. е. вне зависимости от того, сколько бензина осталось в баке машины, на заправке доливается бензин до полного бака, и за это платится фиксированная сумма денег). Машина может (но не обязана) заправиться только в том случае, если её бак заполнен менее чем наполовину. Необходимо определить самый дешёвый маршрут из города  $A$  в город  $B$ , если первоначально бак машины полон.

#### Формат входных данных

В первой строке записаны два целых числа  $N$  ( $2 \leq N \leq 300$ ) и  $M$  ( $1 \leq M \leq 5000$ ). Во второй строке заданы два целых числа  $Z$  ( $1 \leq Z \leq 10000$ ) и  $X$  ( $1 \leq X \leq 100$ ). В третьей строке указаны числа  $A$  и  $B$  ( $1 \leq A, B \leq N$  и  $A \neq B$ ). Далее строка из  $N$  неотрицательных целых чисел,  $i$ -е число — стоимость  $S_i$  дозаправки ( $1 \leq S_i \leq 10000$ ) или число 0, если в этом городе заправки нет. Далее  $M$  строк занимают описания каждой дороги: первые два числа — номера городов, связанных двусторонней дорогой, и третье число — её протяжённость в километрах (целое число от 1 до 10 000).

#### Формат выходных данных

Первая строка выходного файла должна содержать слово Yes, если решение существует, или слово No, если оно отсутствует. Если решение найдено, то во второй строке должны следовать номера городов в порядке их посещения, начиная с номера города  $A$  и заканчивая номером города  $B$ . Если вы заправляетесь в каком-либо городе, в котором есть заправка, то номер этого города нужно вывести со знаком минус. Если решение неоднозначно, выведите



любое из них.

**Пример**

входной файл	выходной файл
4 5 10 2 1 4 9 0 7 0 1 3 2 1 2 2 2 3 1 3 4 5 1 4 7	Yes 1 2 -3 4

**Задача 23. Отрезки**

Пусть на плоскости с евклидовой метрикой задано  $N$  красных и  $N$  синих точек (красные точки имеют номера от 1 до  $N$ , а синие — от  $N + 1$  до  $2N$ ). Существует  $N$  отрезков, соединяющих эти точки таким образом, что каждый отрезок соединяет точки различного цвета и каждая точка — концевая только для одного отрезка. Необходимо определить, является ли заданное соединение минимальным по длине (сумма длин всех отрезков) среди всех возможных соединений, удовлетворяющих заданным свойствам.

**Формат входных данных**

В первой строке находится натуральное число  $N$  ( $1 \leq N \leq 100$ ). Следующие  $N$  строк файла содержат по два целых числа: координаты красных точек (в первой строке — координаты красной точки с номером 1, во второй — с номером 2 и т. д.). Затем идут  $N$  строк файла, в каждой из которых указаны два целых числа: координаты синих точек (первая из этих строк — координаты синей точки с номером  $N + 1$ , вторая — с номером  $N + 2$  и т. д.). Каждая из следующих  $N$  строк файла задаёт отрезки соединения и содержит по два числа, определяющих начальные и конечные номера точек отрезков. Координаты красных и синих точек по модулю не превосходят 1000.

**Формат выходных данных**

Выведите в единственной строке сообщение Yes, если заданное соединение является минимальным по длине. Если оно таковым не является, то выведите сообщение No.

**Пример**

входной файл	выходной файл
3 0 1 0 2 0 3 1 3 1 2 1 1 1 6 2 4 3 5	No
1 100 100 200 200 1 2	Yes

**Задача 24. Трубопроводы**

Имеется сеть трубопроводов по перекачке нефти, при этом у участков трубопроводов есть пропускная способность (тонн/час) и стоимость транспортировки (транзита) тонны нефти по участку. Необходимо организовать перекачку максимального количества тонн нефти при минимальных суммарных затратах на транзит из пункта  $A$  в пункт  $B$ . Если два пункта  $X$  и  $Y$  связаны трубопроводом, то по этому трубопроводу можно перекачивать нефть как из  $X$  в  $Y$ , так и из  $Y$  в  $X$ .

**Формат входных данных**

В первой строке находятся четыре натуральных числа, задающих сеть трубопроводов: чис-



ло  $N$  пунктов, число  $M$  трубопроводов, номера пунктов  $A$  и  $B$  ( $1 < N \leq 100$ ,  $0 \leq M \leq N(N-1)/2$ ). Каждая из следующих  $M$  строк содержит по четыре целых числа, задающих информацию о некотором трубопроводе: два номера пунктов, связанных трубопроводом, его пропускная способность  $P_i$  и стоимость  $S_i$  транспортировки по нему ( $0 \leq P_i \leq 100$ ,  $0 \leq S_i \leq 1000$ ).

#### Формат выходных данных

В первой строке выведите максимальное количество тонн нефти, которое можно перекачать из пункта  $A$  в пункт  $B$ . Во второй строке — минимальные затраты на транзит.

#### Пример

входной файл	выходной файл
6 10 1 6 1 2 4 2 1 3 6 3 2 3 1 5 2 4 3 1 2 5 1 4 3 5 3 2 3 4 2 3 4 5 1 1 4 6 5 3 5 6 6 1	9 60

#### Задача 25. Таксист

Имеется  $N$  городов, связанных  $M$  дорогами (нумерация дорог и городов начинается с единицы). Движение по дорогам осуществляется только в одном направлении, и дороги пересекаются только в городах. Известна длина каждой дороги. Необходимо найти все дороги, по которым потенциально может проехать таксист таким образом, чтобы длина его маршрута отличалась не более чем на величину  $K$  от длины кратчайшего маршрута из 1-го города в  $N$ -й.

#### Формат входных данных

Первая строка файла содержит числа  $N$ ,  $M$  и  $K$  ( $2 \leq N \leq 10000$ ,  $1 \leq M \leq 100000$ ,  $0 \leq K \leq 10000$ ). Каждая из последующих  $M$  строк описывает дорогу: начальный город, конечный город и её длина (целое число не более 10 000).

#### Формат выходных данных

Первая строка содержит число  $L$  дорог, которые таксист потенциально может использовать. Каждая из следующих  $L$  строк — одно число: номер потенциальной дороги. Номера дорогам присваиваются в соответствии с тем, в какой последовательности они были введены, а сами дороги следуют в порядке возрастания их номеров.

#### Пример

входной файл	выходной файл
4 5 1 1 2 1 1 3 4 2 3 1 2 4 3 3 4 1	4 1 3 4 5

#### Задача 26. Стрельба

На соревнованиях по стрельбе каждый участник будет стрелять в цель, которая представляет собой прямоугольник, разделённый на квадраты. Цель содержит  $R \cdot C$  квадратов, расположенных в  $R$  строках и  $C$  столбцах. Квадраты выкрашены в белый или чёрный цвет. В каждом столбце находится ровно 2 белых и  $R-2$  чёрных квадрата. Строки пронумерованы от 1 до  $R$  сверху вниз, а столбцы — от 1 до  $C$  слева направо. Стрелок имеет ровно  $C$  стрел. Последовательность из  $C$  выстрелов называется корректной, если в каждом столбце поражён ровно один белый квадрат, а в каждой строке — не менее одного белого квадрата. Необходимо проверить, существует ли корректная последовательность выстрелов, и если да, то найти одну из них.

### Формат входных данных

Первая строка содержит два целых числа  $R$  и  $C$  ( $2 \leq R \leq C \leq 1000$ ). Эти числа определяют количество строк и столбцов. Каждая из следующих  $C$  строк в блоке содержит два натуральных числа, разделённых пробелом. Числа в  $(i+1)$ -й строке определяют номера строк, где расположены белые квадраты в  $i$ -м столбце.

### Формат выходных данных

Выведите последовательность из  $C$  чисел, соответствующих корректной последовательности выстрелов:  $i$ -й элемент последовательности — номер строки, белую клетку которой поразил выстрел  $i$ . Если такой последовательности не существует, то выведите сообщение No.

Пример

входной файл	выходной файл
4 4 2 4 3 4 1 3 1 4	2 3 1 4
3 3 2 3 2 3 2 3	No

### Задача 27. Инкассаторы

Когда Макс Крейзи закончил финансовый колледж, он стал управляющим городского банка. Уже с первых дней работы Макс столкнулся с одной неразрешимой для него проблемой. В стране, где живёт Макс, есть  $N$  городов. Некоторые из них связаны двусторонними дорогами, пересекающимися только в городах. Раз в месяц инкассаторы Макса должны доставить деньги в  $K$  сберкасс. Все  $K$  сберкасс находятся в разных городах. Банк Макса небогат и имеет одну машину для перевозки денег. Вам необходимо помочь Максиму составить маршрут минимальной длины, который начинается в городе  $L$  (в этом городе находится банк Макса), проходит по всем  $K$  городам, где располагаются нужные сберкассы, и заканчивается также в городе  $L$ .

### Формат входных данных

В первой строке через пробел указаны три числа:  $N$ ,  $M$  и  $K$ , где  $N$  — общее число городов в стране,  $M$  — общее число двусторонних дорог,  $K$  — число сберкасс, которые должны посетить инкассаторы ( $1 < N \leq 100$ ,  $0 < M \leq N(N-1)/2$ ,  $0 < K < 18$ ,  $K < N$ ). Во второй строке через пробел идут  $K$  чисел — номера городов, где находятся сберкассы. Следующие  $M$  строк содержат описание дорог. В каждой строке тремя целыми числами  $X$ ,  $Y$  и  $S$  описывается одна дорога, причём  $X$  и  $Y$  — номера городов, связанных дорогой, и  $S$  — длина этой дороги ( $0 < S \leq 50000$ ,  $1 \leq X, Y \leq N$ ). В последней строке дано число  $L$  — номер города, где находится банк Макса ( $1 \leq L \leq N$ ).

### Формат выходных данных

Единственная строка должна содержать набор чисел — номеров городов искомого маршрута в порядке обхода. В случае когда такого маршрута не существует, выведите одно слово — NO.

### Примеры

входной файл	выходной файл
<pre> 4 6 3 2 4 3 1 2 10 2 4 3 1 3 5 1 4 4 3 2 7 4 3 8 1 3 1 1 3 1 2 100 2 </pre>	<pre> 1 3 2 4 1 NO </pre>

### Задача 28. Город С

Задана система дорог, определяемая набором пар городов. Каждая такая пара  $\{i, j\}$  указывает, что из города  $i$  можно проехать в город  $j$  и в обратном направлении. Необходимо определить, можно ли проехать из заданного города  $A$  в заданный город  $B$  таким образом, чтобы посетить город  $C$ , не проезжать ни по какой дороге более одного раза и не заезжать ни в какой город более одного раза.

#### Формат входных данных

В первой строке находится число  $N$  городов (города нумеруются от 1 до  $N$ ,  $1 \leq N \leq 5000$ ). Во второй строке — целое число  $M$  дорог ( $0 \leq M \leq 10000$ ). Далее в каждой из  $M$  строк — пара номеров городов, соединённых некоторой дорогой. В последней,  $(M + 3)$ -й, строке находятся номера городов  $A$ ,  $B$  и  $C$ .

#### Формат выходных данных

В единственной строке выведите сообщение Yes, если искомая цепь существует, или No в противном случае.

### Примеры

входной файл	выходной файл
<pre> 3 2 1 2 2 3 1 3 2 </pre>	Yes
<pre> 3 2 1 3 2 3 1 3 2 </pre>	No
<pre> 6 6 1 3 2 3 3 5 3 4 5 6 4 6 1 2 6 </pre>	No

### Задача 29. Расселение

Руководство фирмы ООО «Вектор», состоящей из двух отделов, решило на празднование годовщины основания организовать поездку всех сотрудников в санаторий. Санаторий располагает одноместными и двухместными номерами. Для укрепления корпоративного духа было принято решение селить в двухместные номера сотрудников из разных отделов. Если в одном из отделов больше людей, чем в другом, то тех, кому не хватило пары, расселяют в одноместные номера. В силу разного возраста сотрудников вводится показатель недовольства, равный разности в возрасте заселяемых в один номер (вычисляется для каждого номера, а не сотрудника). Для тех, кто попадает в одноместный номер, он равен половине возраста.

### Формат входных данных

В первой строке задаются возрасты сотрудников первого отдела, разделённые пробелами. Во второй строке — возрасты сотрудников второго отдела. Возраст сотрудника должен лежать в интервале от 18 до 60.

### Формат выходных данных

Единственная строка должна содержать наименьший суммарный показатель недовольства (вычисляется как сумма показателей всех номеров). Число должно иметь одну цифру после точки.

Пример

входной файл	выходной файл
20 24 18 42	20.0

### Задача 30. Сеть дорог

В Байтленде было решено построить сеть автомобильных дорог с односторонним движением. Каждая дорога должна соединять только два города. Известно, какие дороги возможно построить, а также пропускная способность для каждой из них и стоимость одной единицы пропускной способности по конкретной дороге (удельная стоимость). Задача строителей — построить дороги так, чтобы максимизировать пропускную способность между городами  $X$  и  $Y$  и затратить на это минимальную сумму денег. Первоначально предложен некоторый вариант строительства дорог. Определить, оптимален ли данный выбор.

### Формат входных данных

В первой строке находятся четыре целых числа, задающих сеть дорог:  $N$  — число пунктов,  $M$  — число дорог, начальный и конечный города  $X$  и  $Y$ . Каждая из следующих  $M$  строк содержит по четыре числа, задающих информацию о дороге, которую можно построить: номера начального и конечного городов дороги, её пропускная способность и стоимость одной единицы пропускной способности по данной дороге. Остальные строки содержат информацию об одном из возможных вариантов сети дорог: номеров начального и конечного городов, которые соединяет дорога.

### Формат выходных данных

Если предложенная сеть дорог является оптимальной, то в первой строке выведите Yes; во второй — максимальную пропускную способность между городами  $X$  и  $Y$ ; в третьей — минимальную стоимость, которая будет заплачена за провоз полученного значения пропускной способности. Если предложенная сеть дорог не является оптимальной, то в первой строке выведите No; во второй — максимальную пропускную способность между городами  $X$  и  $Y$  в оптимальной сети дорог; в третьей — минимальную стоимость, которая будет заплачена за провоз полученного значения пропускной способности для оптимальной сети дорог; в четвёртой — максимальную пропускную способность между городами  $X$  и  $Y$  в предложенной сети дорог; в пятой строке — минимальную стоимость, которая будет заплачена за провоз полученного значения пропускной способности для предложенной сети дорог.

**Пример**

входной файл	выходной файл
6 8 1 6 1 2 14 10 2 3 11 6 2 4 5 20 2 6 3 15 3 4 5 9 3 5 6 9 4 6 6 8 5 6 6 9 1 2 2 3 2 4 2 6 3 5 4 6 5 6	No 14 444 14 469

**Задача 31. Боеприпасы**

На складе хранится  $N$  единиц боеприпасов. Было решено их уничтожить в течение  $N$  дней по единице в день. Для каждой единицы боеприпаса известно, в какие дни она может быть уничтожена, и задана степень риска при уничтожении этой единицы в указанный день. Требуется уничтожить все боеприпасы так, чтобы суммарный риск был минимальным.

**Формат входных данных**

В первой строке задаётся число дней  $N$ , равное числу боеприпасов ( $1 \leq N \leq 200$ ). В следующих  $N$  строках — данные о риске уничтожения боеприпасов: в каждой строке сначала задаётся число  $K_i$  ( $1 \leq K_i \leq N$ ) — число дней, в которые соответствующая единица боеприпасов может быть уничтожена. Затем идут  $2K_i$  чисел, описывающих риск уничтожения этой единицы боеприпасов: на нечётных позициях — номера дней от 1 до  $N$ , а на чётных — соответствующие им степени риска (целые числа в пределах от 0 до 200).

**Формат выходных данных**

В первой строке выведите минимальную суммарную степень риска уничтожения всех боеприпасов. Во второй строке —  $N$  чисел:  $i$ -е число означает, какой боеприпас в  $i$ -й день должен быть уничтожен. Если же все боеприпасы уничтожить невозможно, то в единственной строке выведите -1.

**Пример**

входной файл	выходной файл
3 2 2 1 3 4 2 1 7 3 10 2 1 1 2 4	12 3 1 2

**Замечание** Во второй день с риском 1 или в третий день с риском 4 можно уничтожить боеприпас 1. В первый день с риском 7 или в третий день с риском 10 можно уничтожить боеприпас 2. В первый день с риском 1 или во второй день с риском 4 можно уничтожить боеприпас 3. Безопаснее всего уничтожать боеприпасы в порядке 3, 1, 2 с суммарным риском 12.

**Задача 32. Экскурсия**

Экскурсия по Национальному парку Байтленда заключается в следующем: имеются острова, соединённые мостами. Группа собирается в начальном пункте, а затем передвигается по мостам до конечного пункта, причём каждый человек может идти по индивидуальному маршруту. Поскольку разные люди при этом осматривают различные достопримечательности, то стоимость экскурсии также отличается. По каждому из мостов принято ходить только в одном направлении, чтобы посетители не мешали друг другу. Известна некоторая стоимость прохождения каждого моста для одного человека, а стоимость экскурсии для человека равна сумме стоимостей для мостов, которые он собирается пройти. Группа является очень дисци-

планированной, поэтому если кто-то попадает на остров, то он ждёт, пока не соберутся все, кто должен пройти через этот остров. Для каждого моста также известно максимальное число человек, которое он может одновременно принять по технике безопасности. Если мост соединяет острова  $A$  и  $B$ , то можно выбрать любое из направлений с  $A$  на  $B$  или с  $B$  на  $A$ , но только одно. Так, если хотя бы один из посетителей прошёл по мосту в направлении с  $A$  на  $B$ , то и все остальные, кто будет проходить по этому мосту, должны проходить этот мост в этом направлении. Составьте такой план экскурсии из начального пункта в конечный, при котором группа была бы максимальной, а общая стоимость экскурсии для всей группы — минимальной из всех возможных при данном размере группы.

#### Формат входных данных

Первая строка содержит три целых числа, включающих общую информацию о системе мостиков:  $N$  — число островов ( $2 \leq N \leq 30$ ),  $S$  — номер острова, откуда начинаются все экскурсии ( $1 \leq S \leq N$ ),  $F$  — номер острова, на котором группа должна собраться после экскурсии ( $1 \leq F \leq N$ ). Гарантируется, что начальный и конечный острова не совпадают. Каждая из следующих строк содержит по четыре целых числа, задающих информацию о каждом мостике: номера соединённых этим мостиком островов (от 1 до  $N$ ), максимальное число  $M$  человек, которые могут пройти по данному мосту ( $0 \leq M \leq 50$ ), и стоимость  $C$  его прохождения ( $0 \leq C \leq 1000$ ). Между парой островов существует не более одного моста. Никакой мост не соединяет остров с самим собой.

#### Формат выходных данных

Первая строка должна содержать максимальное число человек, которые смогут пройти по парку. Вторая строка — минимальную стоимость этой экскурсии.

#### Примеры

входной файл	выходной файл
<pre> 7 1 7 1 2 10 43 1 3 7 34 1 4 20 35 2 3 14 42 2 5 14 36 2 6 18 39 3 4 9 45 3 5 12 29 4 6 20 48 4 7 19 40 5 6 17 47 6 7 37 39 </pre>	<pre> 37 3800 </pre>
<pre> 3 1 3 2 1 1 1 2 3 1 1 1 3 1 1 </pre>	<pre> 2 3 </pre>

### Задача 33. Машина времени

Между  $N$  населёнными пунктами совершаются пассажирские рейсы на машинах времени. В момент времени 0 вы находитесь в пункте  $A$ . Вам дано расписание рейсов. Требуется оказаться в пункте  $B$  как можно раньше (т. е. в наименьший возможный момент времени). При этом разрешается делать пересадки с одного рейса на другой. Если вы прибываете в некоторый пункт в момент времени  $T$ , то можете уехать из него любым рейсом, который отправляется из этого пункта в момент времени  $T$  или позднее (но не раньше).

#### Формат входных данных

Первая строка содержит целое число  $N$  ( $1 \leq N \leq 1000$ ) населённых пунктов. Вторая строка — два числа  $A$  и  $B$  — номера начального и конечного пунктов. Третья строка — число  $K$  рейсов ( $0 \leq K \leq 1000$ ). Следующие  $K$  строк содержат описания рейсов, по одному на строке. Каждое описание представляет собой четвёрку целых чисел. Первое число каждой четвёрки задаёт номер пункта отправления, второе — время отправления, третье — пункт

назначения, четвёртое — время прибытия. Номера пунктов — натуральные числа в диапазоне от 1 до  $N$ . Пункт назначения и пункт отправления могут совпадать. Время измеряется в некоторых абсолютных единицах и задаётся целым числом, по модулю не превышающим 109. Поскольку рейсы совершаются на машинах времени, то время прибытия может быть как больше времени отправления, так и меньше или равным ему. Гарантируются такие входные данные, что добраться из пункта  $A$  в пункт  $B$  всегда можно.

#### Формат выходных данных

Выведите минимальное время, в которое вы сможете оказаться в пункте  $B$ .

#### Пример

входной файл	выходной файл
2 1 1 2 1 1 2 10 1 10 1 9	0
1 1 1 3 1 3 1 -5 1 -5 1 -3 1 -4 1 -10	-10

#### Задача 34. Максимальное число коней

Задана шахматная доска размера  $N \times M$ , часть клеток которой могут занимать чёрные фигуры. Необходимо на свободные клетки доски расставить максимальное число белых коней (количество белых коней не ограничено), чтобы они не били друг друга.

#### Формат входных данных

В первой строке находится число  $N$  строк доски, во второй строке — число  $M$  столбцов (1  $N, M$  200). В каждой из последующих строк — по два числа  $x$  и  $y$ , разделённых пробелом, которые являются координатами чёрной фигуры ( $x$  — номер строки,  $y$  — столбца). Координаты верхнего левого угла доски — (1, 1).

#### Формат выходных данных

Выведите максимальное число белых коней.

#### Пример

входной файл	выходной файл
7 7 1 1 2 2 4 4 5 6	23

#### Задача 35. Военный поход

В одном феодальном государстве средневековой Европы было  $N$  городов и  $M$  дорог, каждая из которых соединяла некоторые два города. Каждая дорога принадлежала правителю одного из городов (не обязательно из тех, которые она соединяла). Однажды правитель города  $S$  решил объявить войну правителю города  $T$ . Перед ним сразу же встала нелёгкая задача: как довести армию до города  $T$ . Правитель каждого города требует плату за проход войск через его город. Кроме того, правитель города  $S$  может перемещать войска только по дорогам, которые принадлежат ему. При этом он может купить любую дорогу у её владельца за определённую плату (даже если владелец — правитель города  $T$ ). К сожалению, все деньги из казны города  $S$  были потрачены на предыдущий неудачный военный поход, поэтому сначала правителю придётся продать некоторые свои дороги (разумеется, после этого он не сможет провести по ним войска). Помогите правителю выяснить, какие дороги следует продать, а какие купить, чтобы довести войска от города  $S$  до города  $T$  и оплатить проход через все промежуточные города. Все операции продажи и покупки дорожно осуществить до начала



похода, пока правители других городов не догадались о воинственных намерениях правителя города  $S$ .

#### Формат входных данных

Первая строка содержит целые числа  $N$  и  $M$  — количество городов и дорог соответственно ( $2 \leq N \leq 2000$ ,  $1 \leq M \leq 50000$ ). Города нумеруются от 1 до  $N$ , города  $S$  и  $T$  имеют номера 1 и  $N$  соответственно. В каждой из следующих  $N$  строк находится по одному целому числу  $R_i$  — плата за проезд через город  $i$  ( $0 \leq R_i \leq 10000$ ,  $R_1 = R_N = 0$ ). Следующие  $M$  строк содержат описания дорог. Дорога описывается четырьмя целыми числами:  $A_i$ ,  $B_i$ ,  $P_i$  и  $C_i$ , где  $A_i$  и  $B_i$  — номера городов, которые соединяет дорога ( $A_i = B_i$ );  $P_i$  — номер города, правителю которого она принадлежит;  $C_i$  — её стоимость ( $1 \leq C_i \leq 10000$ ). По дороге можно перемещаться в обоих направлениях. Любые два города соединены не более чем одной дорогой.

#### Формат выходных данных

В первой строке выведите список дорог, необходимых для продажи, в следующем формате: сначала число дорог, а затем их номера. Дороги нумеруются с единицы в том порядке, в котором они заданы на входе. Во второй строке выведите список дорог, необходимых для покупки, в том же формате. В третьей строке выведите маршрут похода — номера городов в порядке следования войска. Если решений несколько, выведите любое. Если решения нет, выведите число -1.

#### Пример

входной файл	выходной файл
3 3 0 1 0 1 2 1 10 2 3 1 10 3 1 2 2	1 1 1 3 1 3

#### Задача 36. План эвакуации

В городе было некоторое число муниципальных зданий и убежищ, построенных специально для укрытия муниципальных работников в случае ядерной войны. Каждое убежище имеет ограниченную вместимость. В идеале все рабочие из муниципального здания должны бежать в ближайшее убежище. Однако это приведёт к переполнению некоторых убежищ, в то время как другие будут заполнены частично. Для решения проблемы городской совет разработал специальный план эвакуации. Муниципальным зданиям выделили убежища, указав число рабочих из каждого здания, которые должны использовать данное убежище, и оставили задачу индивидуального назначения руководству здания. План учитывает число рабочих в каждом здании — все они назначены в убежища, а также ограниченную вместимость каждого убежища — в каждое убежище назначено не больше рабочих, чем оно может вместить. Городской совет заявляет, что их план оптимален, так как минимизирует суммарное время, необходимое рабочим, чтобы добраться до назначенных им убежищ. Мэр города, известный своей постоянной враждой с городским советом, нанял вас как независимого консультанта проверить план эвакуации. Ваша задача — либо убедиться, что план действительно оптимален, либо доказать обратное, предоставив другой план с меньшим общим временем достижения убежищ, ясно показав таким образом некомпетентность совета. Собирая начальные сведения по своей задаче, вы выяснили, что город представляет собой прямоугольную сетку. Положение муниципальных зданий и убежищ определяется двумя целыми числами, а время перехода между муниципальным зданием в точке  $(X_i, Y_i)$  и убежищем в точке  $(P_j, Q_j)$  равно  $D_{ij} = |X_i - P_j| + |Y_i - Q_j| + 1$  мин.

#### Формат входных данных

Первая строка содержит два числа  $N$  и  $M$  — число муниципальных зданий в городе (все они пронумерованы от 1 до  $N$ ) и число убежищ (все они пронумерованы от 1 до  $M$ ) соответственно ( $1 \leq N, M \leq 100$ ). Следующие  $N$  строк описывают муниципальные здания. Каждая строка



содержит целые числа  $X_i, Y_i$  и  $B_i$ , где  $X_i, Y_i$  — координаты зданий ( $-1000 \leq X_i, Y_i \leq 1000$ ), а  $B_i$  — число рабочих в соответствующем здании ( $1 \leq B_i \leq 1000$ ). Следующие  $M$  строк описывают убежища. Каждая строка содержит целые числа  $P_j, Q_j$  и  $C_j$ , где  $P_j, Q_j$  — координаты убежищ ( $-1000 \leq P_j, Q_j \leq 1000$ ), а  $C_j$  — вместимость убежища ( $1 \leq C_j \leq 1000$ ). Затем следуют  $N$  строк. Каждая строка описывает план эвакуации одного здания в том порядке, в котором они даны в описании города. План эвакуации  $i$ -го муниципального здания состоит из  $M$  целых чисел  $E_{ij}$ , где  $E_{ij}$  — число рабочих, которые должны эвакуироваться из здания  $i$  в убежище  $j$  ( $0 \leq E_{ij} \leq 1000$ ). Гарантируется, что план корректен, т. е. он позволяет эвакуировать из каждого здания ровно столько рабочих, сколько их там работает, и не превышает вместимость какого-либо убежища.

#### Формат выходных данных

Если план оптимален, выведите слово OPTIMAL. В противном случае выведите слово SUBOPTIMAL в первой строке, а в следующих  $N$  строках опишите свой план в том же формате, что и на входе. Вашему плану не обязательно быть самому по себе оптимальным, но он должен быть корректным и лучше, чем план городского совета.

#### Примеры

входной файл	выходной файл
<pre> 3 4 -3 3 5 -2 -2 6 2 2 5 -1 1 3 1 1 4 -2 -2 7 0 -1 3 3 1 1 0 0 0 6 0 0 3 0 2 </pre>	<pre> SUBOPTIMAL 3 0 1 1 0 0 6 0 0 4 0 1 </pre>
<pre> 1 1 0 0 1 0 0 1 1 </pre>	<pre> OPTIMAL </pre>

Замечание В первом примере план городского совета обеспечивает суммарное время эвакуации 56, а ваш план — 54, поэтому он более оптимален. Во втором примере суммарное время равно 1, план оптимален.

#### Задача 37. Королевское задание

Жил король и имел  $N$  сыновей. И было в королевстве  $N$  красавиц, и король знал, какая из них нравится каждому его сыну. Сыновья были молодые и легкомысленные, так что вполне вероятно, что одному сыну могли нравиться несколько девушек. Король попросил волшебника найти для каждого сына девушку для женитьбы на ней. Волшебник так и сделал — для каждого сына была выбрана девушка, которая ему нравилась и, конечно же, каждая красавица должна была выйти замуж только за одного из сыновей. Однако король посмотрел на список и сказал: «Мне нравится список, что ты сделал, но я не удовлетворён. Для каждого сына я хотел бы знать он женится на любой из этих девушек, любой другой сын должен иметь возможность жениться на девушке, которая ему нравится». Задача для волшебника оказалась слишком сложной. Вы должны спасти голову волшебника, решив эту задачу.

#### Формат входных данных

В первой строке содержится число  $N$  сыновей ( $1 \leq N \leq 2000$ ). Следующие  $N$  строк для каждого из сыновей содержат список девушек, которые ему нравятся: вначале число  $K_i$  девушек, а затем  $K_i$  различных целых чисел от 1 до  $N$ , определяющих девушек. Сумма  $K_i$  не превосходит 200 000. Последняя строка содержит список, сделанный волшебником, из  $N$  различных целых чисел — для каждого из сыновей номер девушки, на которой он мог бы жениться в соответствии со списком. Гарантируется, что список корректен, т. е. каждому сыну нравится девушка, на которой он должен жениться.

### Формат выходных данных

Выведите  $N$  строк. Для каждого сына вначале выведите число  $L_i$  различных девушек, которые ему нравятся, а затем —  $L_i$  различных целых чисел в произвольном порядке, которые соответствуют понравившимся ему девушкам, женившись на любой из которых, он оставляет возможность всем остальным сыновьям жениться на девушках, которые им нравятся.

#### Пример

входной файл	выходной файл
4 2 1 2 2 1 2 2 2 3 2 3 4 1 2 3 4	2 1 2 2 1 2 1 3 1 4
1 1 1 1	1 1

### Задача 38. Платформы

Известная на весь Могилёв компания «Headache» выпустила игру, для которой необходима конструкция, состоящая из маленьких платформ и труб. Платформы разделяются на стартовые (их  $N_1$  штук), финишные ( $N_3$  штук) и промежуточные ( $N_2$  штук). Все стартовые и финишные платформы находятся на одинаковой высоте. Все высоты промежуточных платформ различны. Они меньше высоты стартовых, но больше высоты финишных. Каждой платформе соответствует уникальный номер от 1 до  $N_1 + N_2 + N_3$ . Нумерация следующая: сначала все стартовые платформы, затем промежуточные и, наконец, финишные. Все промежуточные платформы пронумерованы по убыванию высоты, т. е. если номер промежуточной платформы  $A$  меньше номера платформы  $B$ , то высота  $A$  больше высоты  $B$ . На каждой из стартовых платформ находится шарик. Шарик может скатиться с платформы  $A$  на платформу  $B$ , если они соединены трубой и высота  $A$  больше высоты  $B$ . На каждой из финишных платформ может оказаться не более одного шарика. Если шарик находится на некоторой платформе, то игрок может выбрать направление дальнейшего пути шарика, т. е. выбрать платформу, на которую шарик скатится. Для каждой промежуточной платформы задано число  $C$ , равное максимальному количеству шариков, которые могут прокатиться по ней за время игры. Цель игры заключается в том, чтобы на финишных платформах оказалось как можно больше шариков. Вам нужно узнать, какое максимальное число шариков может оказаться на финишных платформах в результате игры.

### Формат входных данных

В первой строке записаны целые числа  $N_1, N_2, N_3$  — соответственно количество стартовых, промежуточных и финишных платформ, где  $0 < N_1, N_3 \leq 50$ ,  $1 < N_1 + N_2 + N_3 \leq 200$ . В последующих  $N_2$  строках для каждой промежуточной платформы  $j$  ( $N_1 + 1 \leq j \leq N_1 + N_2$ ) указано максимальное число шариков  $C_j$ , которые могут прокатиться по платформе за всё время игры ( $0 \leq C_j \leq 50$ ). В каждой из следующих  $N_1 + N_2$  строк для платформы с номером  $i$  ( $1 \leq i \leq N_1 + N_2$ ) сначала указывается число  $K_i$  — количество труб, выходящих из платформы, а затем перечислены  $K_i$  номеров платформ, на которые может скатиться шарик с платформы  $i$ . Гарантируется, что не существует труб как между стартовыми платформами, так и между финишными.

### Формат выходных данных

В первой строке должно находиться единственное число, равное максимальному числу шариков, которые могут оказаться на финишных платформах в результате игры.

**Пример**

входной файл	выходной файл
3 4 3 3 2 1 2 1 4 1 4 1 4 2 5 6 1 7 1 7 3 8 9 10	2

**Задача 39. Безопасные пути**

В некотором далёком царстве есть система дорог, представляющая собой неориентированный граф. Каждая дорога соединяет два города. Между двумя городами — не более одной дороги. Известно, что с течением времени каждая дорога в этом царстве приобрела один из трёх статусов. Первый — статус дороги, на которой стоит царская охрана (далее именуемой  $R_1$ ), второй — статус дороги, на которой регулярно происходят грабежи проезжающих дилижансов (далее именуемой  $R_2$ ), и третий — статус нейтральной дороги ( $R_3$ ). Отсюда видно, что население страны делится на два слоя — добропорядочных граждан и бандитов. Ясно, что обычные граждане ходят только по дорогам типа  $R_1$  и  $R_3$ , а бандиты — по  $R_2$  и  $R_3$ . Поскольку царская казна тратит много денег на содержание данных дорог, то царь решил убрать максимальное число дорог так, чтобы можно было пройти из любого города в любой другой как добропорядочному гражданину, так и грабителю (иначе в стране поднялся бы бунт — взбунтуются или мирные граждане, или бандиты). Необходимо найти максимальное число удаляемых дорог их номера в соответствии с требованиями царя.

**Формат входных данных**

В первой строке записано сначала число  $N$  городов ( $1 \leq N \leq 500$ ), затем число  $M$  дорог в царстве ( $1 \leq M \leq 10000$ ). Затем идут  $M$  троек чисел, описывающих дороги: первые два задают номера городов, которые она соединяет, а третье — статус: 1 —  $R_1$ , 2 —  $R_2$ , 3 —  $R_3$ . Между любыми двумя городами существует не более одной дороги.

**Формат выходных данных**

Выведите сначала число удаляемых дорог, а затем их номера (дороги нумеруются в том порядке, в каком они были заданы во входном файле). Если решения не существует, выдать -1.

**Пример**

входной файл	выходной файл
5 7 1 2 3 2 3 3 3 4 3 5 3 2 5 4 1 5 2 2 1 5 1	2 4 7

**Задача 40. Светофоры**

Дорожное движение в городе Дингилвилле устроено необычным образом. В городе есть перекрёстки и дороги, которыми перекрёстки связаны между собой. Любые два перекрёстка могут быть связаны не более чем одной дорогой. Не существует дорог, соединяющих один и тот же перекрёсток с самим собой. Время проезда по дороге в обоих направлениях одинаково. На каждом перекрёстке находится один светофор, сигнал которого в каждый момент времени может быть либо голубым, либо розовым. Сигнал каждого светофора изменяется периодически: в течение некоторого интервала времени он голубой, а затем в течение другого

интервала — розовый. Движение по дороге между любыми двумя перекрёстками разрешено тогда и только тогда, когда светофоры на обоих перекрёстках этой дороги имеют один и тот же сигнал в момент въезда на эту дорогу. Если транспортное средство прибывает на перекрёсток в момент переключения светофора, то его поведение будет определяться новым сигналом светофора. Транспортные средства могут находиться в состоянии ожидания на перекрёстках. У вас есть карта города, которая показывает время прохождения каждой дороги (целые числа); длительность каждого сигнала для каждого светофора (целые числа); начальный сигнал и оставшееся время действия этого сигнала (целые числа) для каждого светофора. Необходимо определить путь между двумя заданными перекрёстками, позволяющий транспортному средству проехать от начального перекрёстка к конечному за минимальное время с момента старта. Если существуют несколько таких путей, то вы можете вывести любой.

#### Формат входных данных

Первая строка содержит два числа: номер начального и номер конечного перекрёстков. Во второй строке записано число перекрёстков  $N$  ( $2 \leq N \leq 300$ ) и число дорог  $M$  ( $1 \leq M \leq 14000$ ). Перекрёстки нумеруются от 1 до  $N$ . Следующие  $N$  строк содержат информацию об  $N$  перекрёстках. Отдельная строка содержит информацию о перекрёстке с номером  $i$ :  $C_i, R_i, T_i^B, T_i^P$ . Здесь прописная латинская буква  $C_i$  обозначает начальный свет светофора на перекрёстке (В — голубой, Р — розовый). Далее  $R_i$  — оставшееся время действия начального сигнала ( $1 \leq R_i$ ). Числа  $T_i^B$  и  $T_i^P$  — длительность действия голубого и розового сигнала светофора соответственно на перекрёстке  $i$  ( $1 \leq T_i^B, T_i^P \leq 100$ ). Число  $R_i$  не превосходит длительности действия соответствующего сигнала  $C_i$ . Следующие  $M$  строк описывают  $M$  дорог. Каждая строка имеет вид  $i, j, L_{ij}$ , где  $i$  и  $j$  — номера перекрёстков, которые связывает эта дорога,  $L_{ij}$  — время, необходимое для перемещения по дороге ( $1 \leq L_{ij} \leq 100$ ).

#### Формат выходных данных

Если путь существует, то в первой строке выведите минимальное время, необходимое для достижения конечного перекрёстка из начального, а во второй — список перекрёстков, который соответствует найденному минимальному пути. Перекрёстки должны быть записаны в порядке их посещения. Первое число должно быть номером начального перекрёстка, последнее — конечного. Если пути нет, то выведите число 0.

#### Пример

входной файл	выходной файл
1 4 4 5 В 2 16 99 Р 6 32 13 Р 2 87 4 Р 38 96 49 1 2 4 1 3 48 2 3 75 2 4 76 3 4 77	127 1 2 4

#### Задача 41. Достроить дороги

В некотором государстве есть  $N$  городов, которые пронумерованы целыми числами от 1 до  $N$ . Между городами построено  $M$  дорог. Каждая дорога соединяет два города. По дорогам можно путешествовать в обоих направлениях. Определите, какое минимальное число новых дорог нужно достроить, чтобы из каждого города можно было проехать в любой другой.

#### Формат входных данных

В первой строке записано целое число  $N$  — количество городов ( $1 \leq N \leq 100000$ ). Во второй строке — целое число  $M$  — количество уже существующих дорог ( $0 \leq M \leq 100000$ ). В последующих  $M$  строках описаны дороги: пара чисел  $u$  и  $v$ , записанных через пробел, что свидетельствует о том, что между городами  $u$  и  $v$  построена дорога ( $1 \leq u, v \leq N$  и  $u \neq v$ ).

#### Формат выходных данных

Выведите одно число — минимальное количество дорог, добавление которых обеспечит связность дорожной сети в стране.

**Примеры**

входной файл	выходной файл
3 4 1 2 2 3 1 2 3 1	0
3 0	2

#### Задача 42. Надёжная сеть

Акула хочет быть в курсе всех дел Квадратного Бизнесмена. Для этого он поручил Фицджеральду новое задание: связать в сеть компьютеры Акулы и Квадратного Бизнесмена, используя для этого, возможно, промежуточные компьютеры, стоящие в цехах «Сыр Индастриз». Акуле также известно, что Квадратный Бизнесмен имеет достаточно могущества, чтобы отключить от сети одновременно до  $K$  промежуточных компьютеров, но на отключение  $(K + 1)$ -го компьютера его власти уже недостаточно. Квадратный Бизнесмен никогда не выключает свой компьютер и не может отключить компьютер Акулы, ведь они в деле. Акула хочет, чтобы сеть была надёжной, т. е. чтобы Квадратный Бизнесмен не смог прервать связь между компьютерами Акулы и Квадратного Бизнесмена, даже если бы очень этого захотел. Все компьютеры пронумерованы уникальными натуральными числами от 1 до  $N$ . Компьютер Акулы имеет номер  $A$ , а компьютер Квадратного Бизнесмена — номер  $B$ . Фиц уже определил, какие пары компьютеров можно связать между собой непосредственно и сколько метров провода понадобится для этого. Но теперь он решил, что ему нужно зайти в бар к Роде, чтобы немного освежиться. А вы как его Друган должны посчитать, можно ли построить надёжную сеть. Если это возможно, то узнайте наименьшую суммарную длину проводов (в метрах), которая понадобится для этого, если нет — определите минимальное количество компьютеров, которое нужно отключить Квадратному Бизнесмену, чтобы Акула не мог быть в курсе его дел.

##### Формат входных данных

В первой строке входного файла содержатся два целых числа  $N$  и  $M$  ( $1 \leq N, M \leq 10000$ ) — соответственно количество компьютеров и количество пар компьютеров, которые возможно соединить. У каждой из следующих  $M$  строк есть по три целых числа  $X, Y, Z$  ( $1 \leq X, Y \leq N, 1 \leq Z \leq 20000$ ), означающих, что компьютеры  $X$  и  $Y$  можно соединить непосредственно и на это уйдёт  $Z$  метров провода. Последняя строка содержит три целых числа:  $A, B$  и  $K$  ( $1 \leq A, B \leq N, 0 \leq K \leq N$ ).

##### Формат выходных данных

Первая строка выходного файла должна содержать сообщение Yes, если можно построить надёжную сеть, в противном случае — No. Если ответ положительный, то во второй строке выведите одно целое число, равное наименьшей суммарной длине проводов (в метрах), необходимой для построения сети. Иначе во второй строке выведите одно целое число — минимальное количество компьютеров, которое нужно отключить Квадратному Бизнесмену, чтобы Акула не мог быть в курсе его дел.

### Примеры

входной файл	выходной файл
8 11 1 2 1 1 3 1 2 3 7 2 4 6 3 7 3 3 8 4 4 5 9 4 8 1 5 6 5 6 7 3 6 8 2 2 6 2	No 2

### Задача 43. Острова

Одно небольшое малоизвестное государство размещается на  $N$  островах, омываемых бескрайними водами Тихого океана. На трёх островах расположены три больших города — крупные промышленные узлы государства. Правительство разрабатывает проект строительства автомобильных мостов между некоторыми парами островов, чтобы обеспечить надёжную транспортную связь между городами. Цель — сделать возможным беспрепятственный проезд на автомобиле из каждого города в любой другой. Инженеры разработали  $M$  проектов мостов между парами островов и оценили стоимость реализации всех проектов (каждая пара островов рассматривалась не более одного раза, причём строительство кольцевых мостов из острова в него же никого не интересует). Мосты предполагают двустороннее движение транспорта по ним. Строить все мосты дорого, да и нет необходимости: достаточно связать лишь три острова, на которых располагаются города, можно не напрямую, а через другие острова. Какой минимальный объём финансирования потребуется для этого?

#### Формат входных данных

В первой строке входного файла записаны через пробел целые числа  $N$  ( $3 \leq N \leq 100000$ ) и  $M$  ( $1 \leq M \leq 100000$ ) — соответственно количество островов и количество мостов, которые можно построить. Острова имеют номера от 0 до  $N - 1$ . В последующих  $M$  строках описываются мосты: каждая строка содержит три числа  $A$ ,  $B$  ( $0 \leq A, B < N$ ) и  $K$  ( $1 \leq K \leq 1000000$ ), которые свидетельствуют, что строительство моста между островами  $A$  и  $B$  стоит  $K$  у. е. В последней строке записано три различных числа  $X$ ,  $Y$  и  $Z$  ( $0 \leq X, Y, Z < N$ ) — номера островов, между которыми следует организовать сухопутные перемещения.

#### Формат выходных данных

Выведите одно число — минимальную сумму (в у. е.), которую потребуется потратить на строительство мостов. Гарантируется, что решение существует.

### Примеры

входной файл	выходной файл
3 2 0 1 1 1 2 1 0 1 2	2
4 4 0 1 4 1 2 5 1 3 1 3 2 2 0 1 2	7

### Задача 44. Снести

Издавна сложилось, что большинство сельского населения страны Байтландии селилось на хуторах, состоящих из небольшого числа домов. Постройку и поддержание в надлежащем состоянии дорог, соединяющих некоторые из этих хуторов, взяло на себя государство. Однако затея оказалась слишком дорогостоящей. К настоящему времени между некоторыми парами

хуторов может не существовать пути, проходящего по государственным дорогам. Будем считать, что такие хутора находятся в различных изолированных областях. Денег на постройку новых дорог в казне нет. Поэтому королевские советники решили объявить некоторые хутора бесперспективными и снести их, предоставив обитателям благоустроенное жильё в других населённых пунктах. На первом этапе реализации этой программы чиновники предлагают объявить бесперспективными два хутора, расположенных в разных изолированных областях. Однако король перед утверждением этого плана потребовал (наверное, по своей прихоти), чтобы число изолированных областей в Байтеландии не изменилось после сноса бесперспективных хуторов. И теперь советники задались вопросом: какую же пару хуторов снести? Помогите им, рассчитав число вариантов выбора такой пары.

#### Формат входных данных

В первой строке заданы число  $N$  хуторов и число  $M$  дорог в Байтеландии ( $3 \leq N \leq 100000$ ,  $0 \leq M \leq 100000$ ). Каждая из последующих  $M$  строк описывает одну дорогу и содержит номера хуторов, которые соединяет эта дорога. Нумерация хуторов начинается с единицы. Между любой парой хуторов может быть не более одной дороги. Никакая дорога не соединяет хутор сам с собой.

#### Формат выходных данных

Выведите единственное число — искомое число способов выделения пары бесперспективных хуторов.

#### Примеры

входной файл	выходной файл
<pre> 4 2 1 2 3 4 </pre>	4
<pre> 7 6 1 2 2 3 3 1 4 5 5 6 5 7 </pre>	9

#### Задача 45. Сталкер: туда и обратно

Спасаясь от монстров в заброшенных подземельях НИИ «Агропром», вы юркнули в неприметную дверь — и она вдруг закрылась, оставив вас в лабиринте со множеством комнат и приоткрытыми дверями, ведущими из одной комнаты в другую. К счастью, ваш верный лэптоп, как всегда, не подвёл — вы сумели скачать карту лабиринта. Лабиринт, оказывается, состоит из квадратных комнат, образующих прямоугольник. Некоторые комнаты замурованы (там, видно, хранится что-то ценное, но у вас не хватает опыта, чтобы туда проникнуть). Из каждой комнаты можно перейти в комнаты, которые имеют общую сторону (соседние комнаты), естественно, не выходя за пределы лабиринта и не проникая в замурованные комнаты. Единственный выход из лабиринта находится в той комнате, куда вы так неудачно попали (эта комната — крайняя на плане лабиринта). Ключ от двери находится в другой комнате, так что вам предстоит добраться до ключа и вернуться обратно. Одно радует — монстров в этом лабиринте нет. К сожалению, в каждом переходе между любыми двумя комнатами расположены датчики движения, которые намертво закрывают дверь между этими комнатами, если по переходу кто-то прошёл, так что каждым переходом можно воспользоваться не более одного раза. Переход из одной комнаты в соседнюю занимает одну единицу времени, а ресурс вашего противорадиационного костюма не бесконечен. Именно поэтому вы должны выбраться из лабиринта как можно скорее.

#### Формат входных данных

В первой строке входного файла записаны два числа  $M$  и  $N$  — количество строк и столбцов в плане лабиринта ( $1 \leq M, N \leq 50$ ). Далее следуют  $M$  строк, каждая из которых содержит  $N$



. — пустая комната;  
 \* — замурованная комната;  
 $M$  — комната, в которой находится вход и выход из лабиринта;  
 $G$  — комната, в которой находится ключ.

В выходной файл заносится единственное число — минимальное время, необходимое вам для путешествия по переходам лабиринта, чтобы открыть выходную дверь. Если решение задачи по какой-либо причине невозможно, выведите число -1.

входной файл	выходной файл
4 6 M..... .****. .***** ...*.G	18
4 6 M..... .****. .***** ...*.G	-1

У Квадратного Бизнесмена очень много дел в разных городах. Поэтому нередко ему приходится перемещаться из одного города в другой. Сегодня как раз такой случай. Квадратный Бизнесмен предпочитает передвигаться на самолёте, потому что это очень быстро и удобно. Поскольку Квадратный Бизнесмен летает часто, он пронумеровал все города, в которых приходится бывать, целыми числами от 0 до  $N - 1$ . Сегодня Квадратный Бизнесмен находится в городе  $S$ , но на днях он должен оказаться на важном совещании, которое проходит в городе  $T$ . Зная цены билетов на все авиарейсы, найдите наилучший по стоимости перелёт из  $S$  в  $T$ , совершающий не более  $Q$  пересадок (Квадратный Бизнесмен, как и автор задачи, считает очень утомительным большое число пересадок при авиаперелёте). Для примера, если совершается полёт из города 0 в город 4, а затем из города 4 в город 3, то использовано два рейса и сделана одна пересадка.

В первой строке содержатся два целых числа  $N$  и  $M$  — число городов и число рейсов соответственно ( $1 \leq N, M \leq 105$ ). Каждая из следующих  $M$  строк имеет три целых числа  $X$ ,  $Y$  и  $Z$ , означающих, что из города  $X$  в город  $Y$  есть авиарейс стоимостью  $Z$  ( $0 \leq X, Y < N$ ,  $1 \leq Z \leq 1000$ ). Между одной и той же парой городов может быть несколько авиарейсов, возможно, в различных направлениях. Однако нет авиарейса, который ведёт из города в этот же город. Последняя строка содержит три числа  $S$ ,  $T$  и  $Q$  ( $0 \leq S, T, Q < N$ ). Города  $S$  и  $T$  не совпадают. Известно также, что  $N \times Q \leq 5 \cdot 10^6$ ,  $M \times Q \leq 5 \cdot 10^6$ .

В первой строке выведите Yes, если можно удовлетворить требования Квадратного Бизнесмена, в противном случае — No. Если ответ положительный, то во второй строке выведите два целых числа: стоимость  $C$  и число  $R$  пересадок в наилучшем по стоимости перелёте из тех, которые требуют не более  $Q$  пересадок, а в третьей строке через пробел  $R + 2$  целых числа — номера городов, которые посетит Квадратный Бизнесмен на этом пути (в порядке посещения).



### Примеры

входной файл	выходной файл
5 5 0 1 2 1 2 5 2 3 3 0 4 5 4 3 10 0 3 1	Yes 15 1 0 4 3

### Задача 47. Нефть

В некоторой стране имеются  $N$  нефтяных вышек,  $i$ -я из которых способна выкачать до  $a_i$  тысяч баррелей нефти в сутки, и  $M$  нефтеперерабатывающих заводов, каждый из которых, в свою очередь, способен переработать до  $b_j$  тысяч баррелей нефти в сутки. Заводы и вышки связаны трубопроводами; ввиду различных причин стоимость транспортировки одной тысячи баррелей от  $i$ -й вышки к  $j$ -му заводу составляет  $c_{ij}$  денежных единиц; объём перекачки не ограничен. Учитывая, что вышки и заводы должны находиться в производственном балансе, т. е. вся выкачиваемая нефть должна быть вовремя переработана (излишки нигде не должны сливаться или накапливаться), какое максимальное количество нефти в сутки может перерабатывать данная инфраструктура? На данный вопрос ответить относительно легко, поэтому нас интересует лишь величина минимальных возможных затрат среди всех максимальных по объёму перерабатываемой нефти решений.

#### Формат входных данных

В первой строке входного файла находятся целые числа  $N$  и  $M$  ( $1 \leq N, M \leq 300$ ) — число вышек и заводов соответственно. Во второй строке через пробел перечислены  $N$  целых чисел — значения  $a_i$  ( $1 \leq a_i \leq 30000$ ). В третьей строке через пробел даны  $M$  целых чисел — значения  $b_j$  ( $1 \leq b_j \leq 30000$ ). Сумма всех  $a_i$  равна сумме всех  $b_j$ . В последующих  $N$  строках по  $M$  целых чисел записаны стоимости транспортировки одной тысячи баррелей нефти от соответствующей вышки к заводу;  $j$ -е число в  $(i + 2)$ -й строке задаёт  $c_{ij}$  ( $1 \leq c_{ij} \leq 10000$ ).

#### Формат выходных данных

В единственной строке выведите одно целое число — минимальную величину затрат на транспортировку при условии максимизации объёма переработки. Гарантируется, что ответ можно поместить в знаковое тридцатидвухбитное целое.

### Пример

входной файл	выходной файл
3 4 3 6 7 2 5 1 8 1 2 3 4 8 7 6 5 9 12 10 11	110