

## Содержание

<b>Задачи</b>	<b>2</b>
Задача А. Матрица инцидентности [0.5 sec, 256 mb]	2
Задача В. От матрицы смежности к списку ребер [0.5 sec, 256 mb]	3
Задача С. Дерево [0.5 sec, 256 mb]	4
Задача D. Связанность графа [0.5 sec, 256 mb]	5
Задача Е. Компоненты связности [0.5 sec, 256 mb]	6
Задача F. Поиск пути на гриде [0.8 sec, 256 mb]	7
Задача G. Поиск цикла [0.5 sec, 256 mb]	8
Задача H. Мосты и компоненты [0.5 sec, 256 mb]	9
Задача I. Condense 2. Конденсация графа [0.5 sec, 256 mb]	10
Задача J. Points. Точки сочленения [0.5 sec, 256 mb]	11
Задача K. Компоненты вершинной двусвязности [0.5 sec, 256 mb]	12
<b>Бонус</b>	<b>13</b>
Задача L. Unique Topsort [0.25 sec, 256 mb]	13
Задача M. Раскраска в три цвета [0.5 sec, 256 mb]	14
Задача N. Autotourism [1.0 sec, 256 mb]	15

---

В некоторых задачах большой ввод и вывод. Имеет смысл пользоваться супер быстрым вводом-выводом: <http://acm.math.spbu.ru/~sk1/algo/input-output/>

## Задачи

### Задача А. Матрица инцидентности [0.5 sec, 256 mb]

Вершина графа  $u$  называется *инцидентной* ребру  $e$ , если  $u$  является одним из концов ребра  $e$ .

Аналогично, ребро  $e$  называется *инцидентным* вершине  $u$ , если один из концов  $e$  — это вершина  $u$ .

*Матрицей инцидентности* графа  $G = (V, E)$  называется прямоугольная таблица из  $|V|$  строк и  $|E|$  столбцов, в которой на пересечении  $i$ -ой строки и  $j$ -го столбца записана единица, если вершина  $i$  инцидентна ребру  $j$ , и ноль в противном случае.

Дан неориентированный граф. Выведите его матрицу инцидентности.

#### Формат входных данных

В первой строке входного файла заданы числа  $N$  и  $M$  через пробел — количество вершин и рёбер в графе, соответственно ( $1 \leq N \leq 100$ ,  $0 \leq M \leq 10\,000$ ). Следующие  $M$  строк содержат по два числа  $u_i$  и  $v_i$  через пробел ( $1 \leq u_i, v_i \leq N$ ); каждая такая строка означает, что в графе существует ребро между вершинами  $u_i$  и  $v_i$ . Рёбра нумеруются в том порядке, в котором они даны во входном файле, начиная с единицы.

#### Формат выходных данных

Выведите в выходной файл  $N$  строк, по  $M$  чисел в каждой.  $j$ -ый элемент  $i$ -ой строки должен быть равен единице, если вершина  $i$  инцидентна ребру  $j$ , и нулю в противном случае. Разделяйте соседние элементы строки одним пробелом.

#### Примеры

incident.in	incident.out
3 2	1 0
1 2	1 1
2 3	0 1
2 2	1 1
1 1	0 1
1 2	

**Задача В. От матрицы смежности к списку ребер [0.5 сек, 256 mb]**

Простой неориентированный граф задан матрицей смежности, выведите его представление в виде списка ребер.

**Формат входных данных**

Входной файл содержит число  $N$  ( $1 \leq N \leq 100$ ) — число вершин в графе, и затем  $N$  строк по  $N$  чисел, каждое из которых равно 0 или 1 — его матрицу смежности.

**Формат выходных данных**

Выведите в выходной файл список ребер заданного графа. Ребра можно выводить в произвольном порядке.

**Пример**

m2e.in	m2e.out
3	1 2
0 1 1	2 3
1 0 1	1 3
1 1 0	

### Задача С. Дерево [0.5 сек, 256 mb]

Дан неориентированный граф. Проверьте, является ли он деревом.

#### Формат входных данных

В первой строке входного файла заданы через пробел два целых числа  $n$  и  $m$  — количество вершин и рёбер в графе, соответственно ( $1 \leq n \leq 100$ ). В следующих  $m$  строках заданы рёбра;  $i$ -я из этих строк содержит два целых числа  $u_i$  и  $v_i$  через пробел — номера концов  $i$ -го ребра ( $1 \leq u_i, v_i \leq n$ ). Граф не содержит петель и кратных рёбер.

#### Формат выходных данных

В первой строке выходного файла выведите “YES”, если граф является деревом, и “NO” в противном случае.

#### Примеры

tree.in	tree.out
3 2 1 2 1 3	YES
3 3 1 2 2 3 3 1	NO

### Задача D. Связанность графа [0.5 сек, 256 mb]

Дан граф, содержащий  $N$  вершин и  $M$  рёбер ( $1 \leq N \leq 1000, 1 \leq M \leq 7000$ ). Требуется найти наименьшее число рёбер и эти рёбра, которые нужно добавить, чтобы граф стал связным.

#### Формат входных данных

Во входном файле записаны сначала числа  $N$  и  $M$ , затем идёт описание рёбер графа —  $M$  пар чисел, где каждая пара описывает начало и конец ребра.

#### Формат выходных данных

В первую строку вывести единственное число  $K$  — минимальное количество рёбер, которое нужно добавить. В следующих  $K$  строках выведите по 2 числа — начало и конец нового ребра.

edges.in	edges.out
3 1	1
2 1	1 3

### Задача Е. Компоненты связности [0.5 sec, 256 mb]

Вам задан неориентированный граф с  $N$  вершинами и  $M$  ребрами ( $1 \leq N \leq 20\,000$ ,  $1 \leq M \leq 200\,000$ ). В графе отсутствуют петли и кратные ребра.

Определите компоненты связности заданного графа.

#### Формат входных данных

Граф задан во входном файле следующим образом: первая строка содержит числа  $N$  и  $M$ . Каждая из следующих  $M$  строк содержит описание ребра — два целых числа из диапазона от 1 до  $N$  — номера концов ребра.

#### Формат выходных данных

На первой строке выходного файла выведите число  $L$  — количество компонент связности заданного графа. На следующей строке выведите  $N$  чисел из диапазона от 1 до  $L$  — номера компонент связности, которым принадлежат соответствующие вершины. Компоненты связности следует занумеровать от 1 до  $L$  произвольным образом.

#### Пример

connect.in	connect.out
4 2	2
1 2	1 1 2 2
3 4	

### Задача F. Поиск пути на гриде [0.8 sec, 256 mb]

Дано прямоугольное поле  $W \times H$ . Некоторые клетки проходимы, через некоторые ходить нельзя. Из клетки можно ходить в соседние по ребру (слева, справа, сверху, снизу).

Нужно из клетки  $(x_1, y_1)$  найти любой (не обязательно кратчайший, даже не обязательно простой) путь в клетку  $(x_2, y_2)$ .

#### Формат входных данных

На первой строке  $W, H, x_1, y_1, x_2, y_2$  ( $1 \leq x_1, x_2 \leq W \leq 1000, 1 \leq y_1, y_2 \leq H \leq 1000$ ). Далее  $H$  строк, в каждой из которых по  $W$  символов. Символ "." означает, что клетка проходимая, а символ "\*" означает, что по ней ходить нельзя.

Клетки  $(x_1, y_1)$  и  $(x_2, y_2)$  не совпадают и обе проходимы.

#### Формат выходных данных

Если пути не существует, выведите NO.

Иначе выведите YES и последовательность клеток  $(x_i, y_i)$ , в которой первая совпадает с клеткой  $(x_1, y_1)$ , а последняя с клеткой  $(x_2, y_2)$ .

#### Пример

dfsongrid.in	dfsongrid.out
4 2 1 1 4 2 .... ....	YES 1 1 2 1 3 1 4 1 3 1 3 2 4 2
4 2 1 1 4 2 ..*. .*..	NO
4 2 1 1 4 2 ..*. *...	YES 1 1 2 1 2 2 3 2 4 2

### Задача G. Поиск цикла [0.5 sec, 256 mb]

Дан ориентированный невзвешенный граф. Необходимо определить есть ли в нём циклы, и если есть, то вывести любой из них.

#### Формат входных данных

В первой строке входного файла находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000$ ,  $M \leq 100\,000$ ) — количество вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

#### Формат выходных данных

Если в графе нет цикла, то вывести «NO», иначе — «YES» и затем перечислить все вершины в порядке обхода цикла.

#### Примеры

cycle.in	cycle.out
2 2 1 2 2 1	YES 1 2
2 2 1 2 1 2	NO



### Задача Н. Мосты и компоненты [0.5 sec, 256 mb]

Дан неориентированный граф (не обязательно связный). Граф может содержать петли и кратные ребра.

Выведите все компоненты реберной двусвязности графа (максимальные подмножества вершин, такие что подграф на них не теряет связность при удалении любого ребра).

#### Формат входных данных

Первая строка содержит числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 100\,000$ ) — количество вершин и ребер в графе.

Следующие  $m$  строк задают ребра графа.

#### Формат выходных данных

В первой строке выведите количество компонент, в следующих за ней строках выведите сами компоненты, по одной на строку.

Вершины в каждой компоненте должны идти в возрастающем порядке, компоненты нужно вывести в лексикографическом порядке.

#### Примеры

bridges.in	bridges.out
3 2 1 2 2 3	3 1 2 3
3 3 1 2 2 3 3 1	1 1 2 3
2 2 1 2 1 2	1 1 2
7 8 1 5 5 6 1 6 5 4 4 3 4 2 3 2 7 2	3 1 5 6 2 3 4 7

### Задача 1. Condense 2. Конденсация графа [0.5 сек, 256 mb]

Требуется найти количество ребер в конденсации ориентированного графа. Примечание: конденсация графа не содержит кратных ребер.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $n \leq 10\,000$ ,  $m \leq 100\,000$ ). Следующие  $m$  строк содержат описание ребер, по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i$ ,  $e_i$  — началом и концом ребра соответственно ( $1 \leq b_i, e_i \leq n$ ). В графе могут присутствовать кратные ребра и петли.

#### Формат выходных данных

Первая строка выходного файла должна содержать одно число — количество ребер в конденсации графа.

#### Пример

condense2.in	condense2.out
4 4	2
2 1	
3 2	
2 3	
4 3	

**Задача J. Points. Точки сочленения [0.5 sec, 256 mb]**

Дан неориентированный граф. Требуется найти все точки сочленения в нем.

**Формат входных данных**

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $n \leq 20\,000$ ,  $m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

**Формат выходных данных**

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество точек сочленения в заданном графе. На следующей строке выведите  $b$  целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

**Пример**

points.in	points.out
9 12	3
1 2	1
2 3	2
4 5	3
2 6	
2 7	
8 9	
1 3	
1 4	
1 5	
6 7	
3 8	
3 9	

### Задача К. Компоненты вершинной двусвязности [0.5 сек, 256 mb]

Компонентой вершинной двусвязности графа  $\langle V, E \rangle$  называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $n \leq 20\,000$ ,  $m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

#### Формат выходных данных

В первой строке выходного файла выведите целое число  $k$  — количество компонент вершинной двусвязности графа. Во второй строке выведите  $m$  натуральных чисел  $a_1, a_2, \dots, a_m$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты вершинной двусвязности, которой принадлежит  $i$ -е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

#### Примеры

biconv.in	biconv.out
5 6	2
1 2	1 1 1 2 2 2
2 3	
3 1	
1 4	
4 5	
5 1	

## Бонус

### Задача L. Unique Topsort [0.25 sec, 256 mb]

Дан ориентированный ациклический граф  $G$ . Проверить, что существует единственный топологический порядок вершин графа.

#### Формат входных данных

Первая строка входных данных содержит число вершин графа  $n$  ( $1 \leq n \leq 100\,000$ ) и число ребер графа  $m$  ( $0 \leq m \leq 100\,000$ ). Следующие  $m$  строк содержат пары чисел от 1 до  $n$ , задающие начало и конец соответствующего ребра. Гарантируется, что граф не содержит циклов.

#### Формат выходных данных

Если топологический порядок единственный, выведите на первой строке YES, а на второй номера вершин в топологическом порядке, иначе выведите NO.

#### Примеры

unitopsort.in	unitopsort.out
1 0	YES 1
2 1 2 1	YES 2 1
4 2 1 2 4 3	NO

### Задача М. Раскраска в три цвета [0.5 sec, 256 mb]

Петя нарисовал на бумаге  $n$  кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

#### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  — количество кружков и количество линий, которые нарисовал Петя, соответственно ( $1 \leq n \leq 1\,000$ ,  $0 \leq m \leq 20\,000$ ).

Следующая строка содержит  $n$  символов из множества  $\{\text{'R'}, \text{'G'}, \text{'B'}\}$  —  $i$ -й из этих символов означает цвет, в который раскрашен  $i$ -й кружок ( $\text{'R'}$  — красный,  $\text{'G'}$  — зеленый,  $\text{'B'}$  — синий).

Следующие  $m$  строк содержат по два целых числа — пары кружков, соединенных отрезками.

#### Формат выходных данных

Выведите в выходной файл одну строку, состоящую из  $n$  символов из множества  $\{\text{'R'}, \text{'G'}, \text{'B'}\}$  — цвета кружков после перекраски. Если решений несколько, выведите любое.

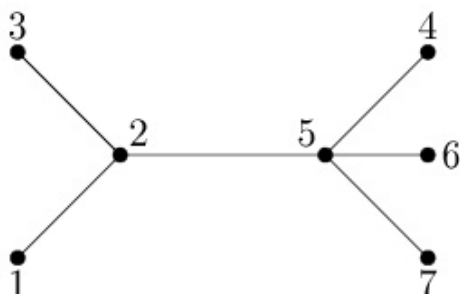
Если решения не существует, выведите в выходной файл слово “Impossible”.

#### Примеры

color.in	color.out
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

### Задача N. Autotourism [1.0 sec, 256 mb]

В Байтландии существуют  $n$  городов, соединённых  $n - 1$  дорогами с двусторонним движением таким образом, что из каждого города можно проехать в любой другой по сети дорог. Длина каждой дороги равна 1 километру.



Бензобак автомобиля позволяет проехать без заправки  $m$  километров. Требуется выбрать маршрут, позволяющий посетить наибольшее количество различных городов без дозаправки. При этом начинать и заканчивать маршрут можно в произвольных городах.

#### Формат входных данных

В первой строке входного файла заданы два целых числа  $n$  и  $m$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq m \leq 200\,000\,000$ ) — количество городов в стране и количество километров, которое автомобиль может проехать без дозаправки. В последующих  $n - 1$  строках описаны дороги. Каждая дорога задаётся двумя целыми числами  $a$  и  $b$  ( $1 \leq a, b \leq n$ ) — номерами городов, которые она соединяет. Длина каждой дороги равна 1 км.

#### Формат выходных данных

Выведите одно число — максимальное количество городов, которое можно посетить без дозаправки.

#### Пример

autotourism.in	autotourism.out
7 6	5
1 2	
2 3	
2 5	
5 6	
5 7	
5 4	

#### Пояснение к примеру

5 городов можно посетить, например, по схеме  $4 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 6 \rightarrow 5 \rightarrow 2$  или по схеме  $3 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 5$ .