

## Содержание

|  |          |
|--|----------|
| <b>Задачи здорового человека</b>                                       | <b>2</b> |
| <b>Задача 10D. Bridges. Мосты [0.3 sec, 256 mb]</b>                    | <b>2</b> |
| <b>Для искателей острых ощущений</b>                                   | <b>3</b> |
| <b>Задача 10E. Мосты и компоненты [0.3 sec, 256 mb]</b>                | <b>3</b> |
| <b>Задача 10F. Points. Точки сочленения [0.3 sec, 256 mb]</b>          | <b>4</b> |
| <b>Задача 10G. Компоненты вершинной двусвязности [0.5 sec, 256 mb]</b> | <b>5</b> |

---

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

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

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

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

---

## Задачи здорового человека

### Задача 10D. Bridges. Мосты [0.3 sec, 256 mb]

Дан неориентированный граф. Требуется найти все мосты 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 мостов в заданном графе. На следующей строке выведите  $b$  целых чисел — номера ребер, которые являются мостами, в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

#### Пример

| stdin | stdout |
|-------|--------|
| 6 7   | 1      |
| 1 2   | 3      |
| 2 3   |        |
| 3 4   |        |
| 1 3   |        |
| 4 5   |        |
| 4 6   |        |
| 5 6   |        |

#### Замечание

Было на лекции. По сути вы просто считаете динамику на остовном дереве `dfs`. Не забывайте рядом с ребром хранить его номер.

## Для искателей острых ощущений

### Задача 10Е. Мосты и компоненты [0.3 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

| stdin   | stdout                   |
|---|--------------------------|
| 3 2<br>1 2<br>2 3   | 3<br>1<br>2<br>3         |
| 3 3<br>1 2<br>2 3<br>3 1                                    | 1<br>1 2 3               |
| 2 2<br>1 2<br>1 2   | 1<br>1 2                 |
| 7 8<br>1 5<br>5 6<br>1 6<br>5 4<br>4 3<br>4 2<br>3 2<br>7 2 | 3<br>1 5 6<br>2 3 4<br>7 |

#### Замечание

Сдайте сперва предыдущую задачу.

Компоненты можно поддерживать на стеке.

### Задача 10F. Points. Точки сочленения [0.3 sec, 256 mb]

Дан неориентированный граф без петель а кратных рёбер. 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

| stdin | stdout |
|-------|--------|
| 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   |        |

#### Замечание

Очень похоже на мосты, чуть другой if «точка сочленения ли».

### Задача 10G. 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

| stdin | stdout      |
|-------|-------------|
| 5 6   | 2           |
| 1 2   | 1 1 1 2 2 2 |
| 2 3   |             |
| 3 1   |             |
| 1 4   |             |
| 4 5   |             |
| 5 1   |             |

#### Замечание

Компоненты живут на стеке.

Компонента вершинной двусвязности — множество рёбер.