

## Задача А. Паросочетание

Имя входного файла: `pairs.in`  
Имя выходного файла: `pairs.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

*Двудольным графом* называется неориентированный граф  $(V, E)$ ,  $E \subseteq V \times V$  такой, что его множество вершин  $V$  можно разбить на два множества  $A$  и  $B$ , для которых  $\forall (e_1, e_2) \in E$   $e_1 \in A$ ,  $e_2 \in B$  и  $A \cup B = V$ ,  $A \cap B = \emptyset$ .

*Паросочетанием* в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор  $S \subseteq E$ , что для любых двух рёбер  $e_1 = (u_1, v_1)$ ,  $e_2 = (u_2, v_2)$  из  $S$   $u_1 \neq u_2$  и  $v_1 \neq v_2$ .

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

### Формат входных данных

В первой строке записаны два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 250$ ), где  $n$  — число вершин в множестве  $A$ , а  $m$  — число вершин в  $B$ .

Далее следуют  $n$  строк с описаниями рёбер —  $i$ -я вершина из  $A$  описана в  $(i+1)$ -й строке файла. Каждая из этих строк содержит номера вершин из  $B$ , соединённых с  $i$ -й вершиной  $A$ . Гарантируется, что в графе нет кратных ребер. Вершины в  $A$  и  $B$  нумеруются независимо (с единицы). Список завершается числом 0.

### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число  $l$  — количество рёбер в максимальном паросочетании. Далее следуют  $l$  строк, в каждой из которых должны быть два целых числа  $u_j$  и  $v_j$  — концы рёбер паросочетания в  $A$  и  $B$  соответственно.

### Пример

| <code>pairs.in</code> | <code>pairs.out</code> |
|-----------------------|------------------------|
| 2 2                   | 2                      |
| 1 2 0                 | 1 1                    |
| 2 0                   | 2 2                    |

## Задача В. Покрытие путями

Имя входного файла: `paths.in`  
Имя выходного файла: `paths.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Задан ориентированный ациклический граф. Требуется определить минимальное количество не пересекающихся по вершинам путей, покрывающих все вершины.

### Формат входных данных

Первая строка входного файла содержит целые числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $2 \leq n \leq 1000$ ,  $0 \leq m \leq 10^5$ ). В следующих  $m$  строках содержатся по два натуральных числа — номера вершин  $u$  и  $v$ , которые соединяет ребро  $(u, v)$ .

### Формат выходных данных

В первой строке выходного файла выведите натуральное число  $k$  — минимальное количество путей, необходимых для покрытия всех вершин.

### Пример

| <code>paths.in</code>    | <code>paths.out</code> |
|--------------------------|------------------------|
| 3 3<br>1 3<br>3 2<br>1 2 | 1                      |

## Задача С. Шахматная доска

Имя входного файла: `chess.in`  
Имя выходного файла: `chess.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася любит играть в необычные шахматы. Его братишка Коля был еще очень маленький. Как-то раз, когда Вася вернулся из школы, он увидел, что его любимую шахматную доску кто-то перекрасил. Вася не сильно разозлился на Колю, потому что очень любил своего брата. Так как у них дома были только черная и белая краски, каждая клетка доски была покрашена в один из этих двух цветов.

Вася решил исправить ошибку брата, он решил покрасить доску так, чтобы она снова стала шахматной. Но Вася почему-то подумал, что хочет красить только диагонали. Причем Вася решил не тратить много времени, поэтому его интересует способ покраски, который содержит наименьшее количество действий. За одно действие Вася может покрасить полностью какую-либо диагональ, в любой из двух цветов: черный или белый. Диагонали бывают двух типов, в зависимости от направления прямой, на которой лежит диагональ. Диагональ, которая лежит на прямой, направленной влево и вниз, является диагональю первого типа, а диагональ, которая лежит на прямой, направленной вправо и вниз, — второго.

Вам предстоит помочь Васе. Задано испорченное Колей шахматное поле. Вам необходимо определить, за какое минимальное количество действий Вася сможет перекрасить доску так, чтобы она стала шахматной.

### Формат входных данных

В первой строке входного файла записаны два целых числа:  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ) — количество строчек и количество столбцов шахматного поля соответственно.

В следующих  $n$  строках записано поле, в каждой строке по  $m$  символов. Каждая строка входного файла описывает одну строку шахматного поля.  $W$  соответствует белой клетке,  $B$  — черной.

### Формат выходных данных

В выходной файл нужно вывести число  $p$ , количество действий, которое потребуется Васе, чтобы его доска снова стала шахматной. В следующих  $p$  строках описаны действия. Каждое действие описано тремя параметрами: тип диагонали, координаты клетки и цвет. Тип диагонали — это число 1 или 2. Координаты клетки — это два целых числа: строка и столбец одной из клеток, которую покрасили этим действием. Цвет — это символ  $W$  или  $B$ , белый и черный соответственно.

### Примеры

| chess.in                 | chess.out    |
|--------------------------|--------------|
| 3 3<br>WBB<br>BWB<br>BBW | 1<br>1 3 1 W |
| 3 3<br>WBW<br>WWB<br>WWW | 1<br>2 2 1 B |

## Задача D. Поток

Имя входного файла: `flow.in`  
Имя выходного файла: `flow.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Вам задан ориентированный граф  $G$ , каждое ребро имеет собственную пропускную способность. Найдите максимальный поток между вершинами 1 и  $n$ .

### Формат входных данных

Первая строка содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер в графе ( $2 \leq n \leq 1000$ ,  $1 \leq m \leq 30\,000$ ). Следующие  $m$  строк описывают ребра:  $u, v, c$  ( $1 \leq u, v \leq n$ ,  $0 \leq c \leq 10^9$ ) — ребро из вершины  $u$  в вершину  $v$  пропускной способностью  $c$ .

### Формат выходных данных

Выведите единственное число — величину потока между вершинами 1 и  $n$ .

### Примеры

| flow.in  | flow.out |
|--|----------|
| 4 5<br>1 2 1<br>1 3 2<br>3 2 1<br>2 4 2<br>3 4 1 | 3        |

## Задача Е. Разрезание графа

Имя входного файла: `mincut.in`  
Имя выходного файла: `mincut.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Разбейте множество вершин заданного графа на два непустых подмножества  $A$  и  $B$  так, чтобы количество рёбер между вершинами различных подмножеств было минимально.

### Формат входных данных

В первой строке входного файла записано целое число  $n$  ( $2 \leq n \leq 50$ ) — число вершин в графе. Каждая из следующих  $n$  строк содержит по  $n$  символов.  $i$ -ый символ  $j$ -ой из этих строк равен “1”, если между вершинами  $i$  и  $j$  есть ребро, и “0” в противном случае. Заданная таким образом матрица смежности графа является антирефлексивной (на главной диагонали стоят нули) и симметричной (относительно главной диагонали).

### Формат выходных данных

Выведите в выходной файл две строки. На первой выведите номера вершин, попавших во множество  $A$ , через пробел, а на второй — номера вершин, попавших во множество  $B$ , также через пробел. Номера вершин можно выводить в любом порядке.

### Примеры

| <code>mincut.in</code>            | <code>mincut.out</code> |
|-----------------------------------|-------------------------|
| 4<br>0111<br>1001<br>1001<br>1110 | 2<br>1 3 4              |

## Задача F. Максимальный поток

Имя входного файла: `maxflow.in`  
Имя выходного файла: `maxflow.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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

### Формат входных данных

Первая строка входного файла содержит  $n$  и  $m$  — количество вершин и количество ребер графа ( $2 \leq n \leq 100$ ,  $1 \leq m \leq 1000$ ). Следующие  $m$  строк содержат по три числа: номера вершин, которые соединяет соответствующее ребро графа и его пропускную способность. Пропускные способности не превосходят  $10^5$ .

### Формат выходных данных

В выходной файл выведите одно число — величину максимального потока из вершины с номером 1 в вершину с номером  $n$ .

### Примеры

| <code>maxflow.in</code>                          | <code>maxflow.out</code> |
|--|--------------------------|
| 4 5<br>1 2 1<br>1 3 2<br>3 2 1<br>2 4 2<br>3 4 1 | 3                        |