

Задача А. От матрицы смежности к списку ребер

Имя входного файла: `m2e.in`
Имя выходного файла: `m2e.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

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

Формат входного файла

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 выходного файла

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

Пример

<code>m2e.in</code>	<code>m2e.out</code>
3	1 2
0 1 1	2 3
1 0 1	1 3
1 1 0	

Задача В. Компоненты связности

Имя входного файла: `matrix.in`
Имя выходного файла: `matrix.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности.

Формат входного файла

В первой строке входного файла содержится одно натуральное число N ($N \leq 100$) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i -ой строке на j -ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

Формат выходного файла

Вывести одно целое число — искомое количество компонент связности графа.

Пример

<code>matrix.in</code>	<code>matrix.out</code>
6 0 1 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0	3

Задача С. Двудольный граф

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

Двудольным называется неориентированный граф $\langle V, E \rangle$, вершины которого можно разбить на два множества L и R , так что $L \cap R = \emptyset$, $L \cup R = V$ и для любого ребра $(u, v) \in E$ либо $u \in L, v \in R$, либо $v \in L, u \in R$.

Дан неориентированный граф. Требуется проверить, является ли он двудольным.

Формат входного файла

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$).

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

Формат выходного файла

В единственной строке выходного файла выведите «YES», если граф является двудольным и «NO» в противном случае.

Пример

bipartite.in	bipartite.out
4 4 1 2 1 3 2 4 4 2	YES
3 3 1 2 2 3 3 1	NO

Задача D. Поиск цикла

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

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

Формат входного файла

В первой строке входного файла находятся два натуральных числа N и M ($1 \leq N \leq 100000$, $M \leq 100000$) — 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 выходного файла

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

Пример

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

Задача Е. Топологическая сортировка

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

Формат входного файла

В первой строке входного файла даны два натуральных числа N и M ($1 \leq N \leq 100\,000$, $0 \leq M \leq 100\,000$) — 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 выходного файла

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести -1.

Пример

<code>topsort.in</code>	<code>topsort.out</code>
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5
3 3 1 2 2 3 3 1	-1

Задача F. Кратчайший путь в невзвешенном графе

Имя входного файла: pathbge1.in
Имя выходного файла: pathbge1.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан неориентированный невзвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Формат входного файла

В первой строке входного файла два числа: n и m ($2 \leq n \leq 30000, 1 \leq m \leq 400000$), где n — количество вершин графа, а m — количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной и конечной вершиной. Вершины нумеруются с единицы.

Формат выходного файла

Выведите n чисел — для каждой вершины кратчайшее расстояние до нее.

Пример

pathbge1.in	pathbge1.out
2 1	0 1
2 1	

Задача G. Получи дерево

Имя входного файла: `graph2tree.in`
Имя выходного файла: `graph2tree.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 16 мегабайт

Дан связный неориентированный граф без петель и кратных ребер. Разрешается удалять из него ребра. Требуется получить дерево.

Формат входного файла

Сначала вводятся два числа: N (от 1 до 100) и M — количество вершин и ребер графа соответственно. Далее идет M пар чисел, задающих ребра. Гарантируется, что граф связный.

Формат выходного файла

Выведите $N - 1$ пару чисел — ребра, которые войдут в дерево. Ребра можно выводить в любом порядке.

Пример

<code>graph2tree.in</code>	<code>graph2tree.out</code>
4 4	1 2
1 2	2 3
2 3	3 4
3 4	
4 1	

Задача H. Island. Островные государства

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

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом, каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город N на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег и помочь ему здесь, как обычно, предлагается Вам.

Формат входного файла

В первой строке входного файла записано два числа N и M ($1 \leq N, M \leq 100\,000$) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах — N чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних M строках записаны пары $1 \leq a, b \leq N, a \neq b$. Каждая пара означает наличие дороги из города a в город b . По дорогам Байтландии можно двигаться в любом направлении.

Формат выходного файла

Если искомого пути не существует, выведите единственное слово `impossible`. В противном случае в первой строке напишите минимальную стоимость и количество посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

Пример

island.in	island.out
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	1 3 1 4 5

Задача I. Площадь комнаты

Имя входного файла: `roomsquare.in`
Имя выходного файла: `roomsquare.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 16 мегабайт

Требуется вычислить площадь комнаты в квадратном лабиринте.

Формат входного файла

В первой строке вводится число N — размер лабиринта ($3 \leq N \leq 10$). В следующих N строках задан лабиринт (‘.’ — пустая клетка, ‘*’ — стенка). И наконец, последняя строка содержит два числа — номер строки и столбца клетки, находящейся в комнате, площадь которой необходимо вычислить. Гарантируется, что эта клетка пустая и что лабиринт окружен стенками со всех сторон.

Формат выходного файла

Требуется вывести единственное число — количество пустых клеток в данной комнате.

Пример

<code>roomsquare.in</code>	<code>roomsquare.out</code>
5 ***** **.* *.*.* *..** ***** 2 4	3

Задача J. Числа

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

С числом разрешается делать одну из двух операций — прибавить один по модулю n или умножить на два по модулю n . Определите, за сколько действий из числа a можно получить число b .

Формат входного файла

В первой строке входного файла даны три натуральных числа n , a и b ($1 \leq n \leq 100\,000$, $0 \leq a < b \leq n$).

Формат выходного файла

В выходной файл выведите одно число — минимальное количество действий.

Пример

<code>numbers.in</code>	<code>numbers.out</code>
4 2 0	1
10 2 9	3