

Задача А. Максимальное паросочетание

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

Граф (V, E) называется *двудольным*, если множество его вершин V можно разбить на два подмножества A и B такие, что любое ребро из E соединяет вершину из A с вершиной из B .

Паросочетанием P называется любое подмножество E такое, что никакие два ребра из него не имеют общей вершины.

Максимальное паросочетание — это паросочетание, число рёбер в котором максимально.

Найдите максимальное паросочетание в заданном двудольном графе.

Формат входного файла

В первой строке входного файла заданы три числа m и n и k ($1 \leq m, n \leq 100$, $1 \leq k \leq 10\,000$), где m — число вершин в множестве A , n — число вершин в B , а k — количество рёбер в графе. Каждая из следующих k строк содержит по два числа u_i и v_i , означающих, что вершина u_i множества A соединена ребром с вершиной v_i множества B . Вершины во множествах A и B нумеруются по отдельности, начиная с единицы. Все числа во входном файле целые.

Формат выходного файла

В первой строке выходного файла выведите число l — количество рёбер в максимальном паросочетании. Далее выведите l строк, по два числа в каждой. Числа a_j и b_j , стоящие в j -ой из этих строк, означают, что в паросочетание взято ребро между вершиной a_j множества A и вершиной b_j множества B . Взятые рёбра должны образовывать максимальное паросочетание.

Пример

bipart.in	bipart.out
2 3 4	2
1 1	1 1
1 2	2 3
2 2	
2 3	

Задача В. Доминошки 2

Имя входного файла: domino2.in
Имя выходного файла: domino2.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Задана прямоугольная доска, некоторые клетки из которой вырезаны. Определите, можно ли покрыть оставшиеся клетки доминошками.

Формат входного файла

В первой строке входного файла заданы два целых числа m и n ($1 \leq m, n \leq 40$) — размеры доски. Каждая из следующих m строк содержит по n символов. i -ый символ j -ой из этих строк равен “X” (икс большое), если клетка вырезана, и “.” (точка), если она пуста.

Формат выходного файла

Выведите в выходной файл “YES”, если доску можно покрыть доминошками, и “NO” в противном случае.

Примеры

domino2.in	domino2.out
2 3 ... XXX	NO
domino2.in	domino2.out
3 2 .X .. X.	YES

Задача С. Максимальный поток

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

Найдите максимальный поток в заданной сети.

Формат входного файла

В первой строке входного файла записаны два числа n и m ($1 \leq n \leq 100$, $1 \leq m \leq 10\,000$) — число вершин и рёбер в сети, соответственно. Каждая из следующих m строк содержит по три числа u_i , v_i и c_i ($1 \leq u_i, v_i \leq n$, $1 \leq c_i \leq 10\,000$),

означающих, что между вершинами u_i и v_i в сети есть ребро с пропускной способностью c_i . Вершина 1 считается истоком, а вершина n — стоком. Граф сети считается неориентированным. Все числа во входном файле целые.

Формат выходного файла

Выведите в выходной файл одно число — максимальный поток в заданной сети.

Пример

flow.in	flow.out
3 3	8
1 2 3	
1 3 5	
3 2 7	

Пример

mincut.in	mincut.out
4	3
0111	1 2 4
1001	
1001	
1110	

Задача D. Разрезание графа

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

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

Формат входного файла

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 выходного файла

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.