

Задача Path. Кратчайший путь

Имя входного файла: path.in
 Имя выходного файла: path.out
 Ограничение по времени: 2 seconds
 Ограничение по памяти: 64 мегабайта

Дан взвешенный ориентированный граф и вершина s в нем. Требуется для каждой вершины u найти длину кратчайшего пути из s в u .

Формат входного файла

Первая строка входного файла содержит n , m и s — количество вершин, ребер и номер выделенной вершины соответственно ($2 \leq n \leq 2000$, $1 \leq m \leq 5000$).

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее 10^{15} по модулю. В графе могут быть кратные ребра и петли.

Формат выходного файла

Выведите n строк — для каждой вершины u выведите длину кратчайшего пути из s в u , '*' если не существует пути из s в u и '-' если не существует кратчайший путь из s в u .

Пример

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

Задача Paths. 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 непересекающихся путей, покрывающих все вершины.

Формат входного файла

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

Формат выходного файла

В первой строке выходного файла выведите натуральное число k — минимальное количество путей, необходимых, чтобы покрыть все вершины.

Примеры

paths.in	paths.out
3 3	1
1 3	
3 2	
1 2	

Задача Dominoes. Замощение доминошками

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

Дано игровое поле размерами $n \times m$, некоторые клетки кото-

рого уже замощены. Замостить свободные соседние клетки поля доминошкой размерами 1×2 стоит a условных единиц. Замостить свободную клетку поля квадратиком размерами 1×1 стоит b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить все поле.

Формат входного файла

Первая строка входного файла содержит 4 числа n , m , a , b ($1 \leq n, m \leq 100$, a, b — целые числа, по модулю не превосходящие 1000). Каждая из последующих n строк содержит по m символов: символ "." (точка) обозначает занятую клетку поля, а символ "*" (звездочка) — свободную.

Формат выходного файла

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

Пример

dominoes.in	dominoes.out
2 3 3 2	5
.*	
.*	

Задача Pairs. 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) , $E \subset V \times V$ 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 $A, B \subset E, A \cap B = \emptyset$.

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

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

Формат входного файла

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 ребер — i -я вершина из A описана в $i + 1$ -й строке файла. Каждая из этих строк содержит номера вершин из B , соединенных с i -й вершиной A . Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

Формат выходного файла

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

Пример

pairs.in	pairs.out
2 2	2
1 2 0	1 1
2 0	2 2