

Задача А. Очередные запросы достижимости

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вам дан простой неориентированный граф с n вершинами и m ребрами.

Вы начинаете в определенной вершине и на каждом ходу должны переходить через ребро к соседней вершине.

Ваша задача — ответить на q запросов вида: «Возможно ли начать с вершины a и закончить в вершине b ровно через x ходов?»

Формат входных данных

Первая строка содержит три целых числа n , m и q ($2 \leq n \leq 2500$, $1 \leq m \leq 5000$, $1 \leq q \leq 10^5$) — количество вершин, ребер и запросов. Вершины пронумерованы $1, 2, \dots, n$.

После этого идут m строк, описывающих ребра. Каждая строка содержит два целых числа a и b ($1 \leq a, b \leq n$) — между вершинами a и b проходит ребро.

Наконец, есть q строк, каждая из которых описывает запрос. Каждая строка содержит три целых числа a , b и x ($1 \leq x \leq 10^9$).

Формат выходных данных

Для каждого запроса выведите ответ («YES» или «NO») в отдельной строке

Пример

стандартный ввод	стандартный вывод
4 5 6	YES
1 2	NO
2 3	YES
1 3	NO
2 4	YES
3 4	YES
1 2 2	
1 4 1	
1 4 5	
2 2 1	
2 2 2	
3 4 8	

Замечание

В запросе 1 возможным маршрутом является $1 \rightarrow 3 \rightarrow 2$.

В запросе 3 возможным маршрутом является $1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4$.

В запросе 6 возможным маршрутом является $3 \rightarrow 4 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4$.

Задача В. Сильносвязная ориентация

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Вам дан простой неориентированный граф из n вершин и m рёбер. Выберите на каждом ребре одну из двух возможных ориентаций так, чтобы результирующий ориентированный граф был сильно связным.

Формат входных данных

Первая входная строка содержит два целых числа n и m ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot 10^5$) — количество вершин и рёбер. Вершины пронумерованы $1, 2, \dots, n$.

После этого идут m строк, описывающих ребра. Каждая строка содержит два целых числа a и b ($1 \leq a, b \leq n$) — между вершинами a и b проходит ребро.

Формат выходных данных

Выведите m строк, описывающих направления ребер.

Каждая строка должна содержать два целых числа a и b — от вершины a до вершины b проходит ребро. Вы можете вывести любое допустимое решение.

Если решений нет, выведите -1 .

Пример

стандартный ввод	стандартный вывод
3 3	1 2
1 2	2 3
1 3	3 1
2 3	

Задача С. Монстры и лабиринт

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Вы и несколько монстров находитесь в прямоугольном лабиринте. Каждую минуту Вы можете сделать шаг в одном из направлений, при этом каждый из монстров также может одновременно с Вами сделать шаг в любом направлении. Ваша цель — добраться до одной из граничных клеток, ни в коем случае не деля клетку с монстром.

Выясните, достижима ли ваша цель, и если да, то выведите путь, по которому вы сможете пройти. Ваш план должен сработать в любой ситуации, даже если монстры знают его заранее.

Формат входных данных

Первая строка ввода содержит два целых числа n и m ($1 \leq n, m \leq 1000$) — высоту и ширину лабиринта.

После этого следуют n строк из m символов, описывающих карту. Каждый символ — это «.» (свободная клетка), «#» (стена), «А» (начало) или «М» (монстр). Во входных данных есть ровно одна буква А.

Формат выходных данных

Сначала выведите «YES», если ваша цель достижима, и «NO» в противном случае.

Если ваша цель достижима, также выведите пример допустимого пути (длина пути и его описание с использованием символов D, U, L и R). Вы можете вывести любой путь, если его длина не превышает $n \cdot m$ шагов.

Пример

стандартный ввод	стандартный вывод
5 8 ##### #M..A..# #.#.M#.# #M#..#.. #.#####	YES 5 RRDDR

Задача D. Классическая задача максимизации

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Вам даны $2n$ различных точек на плоскости. Точка i имеет координаты (x_i, y_i) .

Точки i и j называются *дружной парой* если выполняется $x_i = x_j$ или $y_i = y_j$.

Разбейте точки на n пар. Каждая точка должна принадлежать ровно одной паре. Количество дружных пар среди ваших n пар должно быть наибольшим возможным.

Формат входных данных

Первая строка содержит целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Первая строка каждого набора содержит одно целое число n ($1 \leq n \leq 10^5$).

i -я из следующих $2n$ строк содержит два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты i -й точки. Гарантируется, что все точки различны.

Гарантируется, что сумма n по всем наборам входных данных не превышает 10^5 .

Формат выходных данных

Для каждого набора тестовых данных выведите неотрицательное целое число k — максимальное количество дружных пар.

В i -й из следующих n строк выведите два целых числа a_i и b_i , означающих пару, образованную точками с номерами a_i и b_i ($1 \leq a_i, b_i \leq 2n$; $a_i \neq b_i$).

Каждое целое число от 1 до $2n$ должно быть среди a_i и b_i ровно один раз. Количество индексов i , таких что a_i и b_i образуют дружную пару, должно равняться k .

Пример

стандартный ввод	стандартный вывод
3	2
2	2 4
0 0	3 1
0 1	2
1 0	4 3
1 1	2 1
2	0
0 0	1 2
0 1	3 4
0 2	
0 3	
2	
0 0	
1 1	
2 2	
3 3	

Задача Е. Берендеевы поляны, перестройка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Летняя смена ЛКШ 2026 тоже прошла в Берендеевых полянах, но на этот раз база претерпела многочисленные изменения: во-первых, проложили новые дорожки между домиками, да причём не абы как, а таким образом, чтобы каждая дорожка принадлежала не более чем одному простому циклу, во-вторых, от каждого домика можно дойти до любого другого, в-третьих, все домики перекрасили в три цвета, причём краска третьего типа оказалась очень дорогой. Также, чтобы не было скучно, любые два домика, соединённые дорожкой, должны быть покрашены в разные цвета. Помогите работникам базы покрасить домики в соответствии со всеми условиями, и из всех таких способов найдите самый дешёвый, или же скажите, что такой раскраски в принципе не существует.

Более формально у вас есть n домиков соединённые m двусторонними дорожками. Граф домиков связный и является рёберным кактусом. Вам нужно покрасить вершины в три цвета и минимизировать количество вершин третьего цвета. Также соседние вершины по ребру должны быть покрашены в разные цвета.

Формат входных данных

В первой строке вам дано два числа $1 \leq n \leq 100000$ и $0 \leq m \leq 150000$ – количество домиков и дорожек. В следующих m строках дано по два целых числа a_i и b_i – номера домиков соединённые i -й дорожкой. ($1 \leq a_i, b_i \leq n, a_i \neq b_i$). Гарантируется, что граф связан, является рёберным кактусом и в нём нет кратных рёбер и петель.

Формат выходных данных

Если раскрасить домики требуемым образом невозможно, то выведите -1 , иначе выведите минимально возможное количество домов, которые нужно покрасить в третий цвет.

Примеры

стандартный ввод	стандартный вывод
7 9 5 2 3 6 6 7 7 3 1 2 2 3 3 1 2 4 4 5	2
15 18 13 10 2 14 14 9 7 8 8 3 9 15 15 10 1 2 4 5 5 6 6 7 2 9 2 3 3 4 9 10 10 11 11 12 12 13	1

Задача Ф. Красивые циклы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Мистер Нежмах имеет особую симпатию к m **неупорядоченным** тройкам попарно различных чисел (a_i, b_i, c_i) .

Циклом длины k называется последовательность (необязательно различных) чисел $v_1, v_2, \dots, v_k$ такая, что каждые три подряд идущих числа образуют тройку, которую Нежмах любит, а также тройки (v_{k-1}, v_k, v_1) и (v_k, v_1, v_2) тоже входят в список любимых.

Цикл называется **кривым**, если его длина не делится на 3.

Необходимо проверить, существует ли кривой цикл.

Формат входных данных

Первая строка содержит одно целое число t — количество наборов входных данных ($1 \leq t \leq 2 \cdot 10^4$).

В первой строке каждого набора записано одно целое число m — количество троек ($1 \leq m \leq 2 \cdot 10^5$).

В следующих m строках заданы целые числа a_i, b_i и c_i — элементы тройки ($1 \leq a_i, b_i, c_i \leq 2 \cdot 10^5$). Для каждого i числа a_i, b_i и c_i попарно различны, и все неупорядоченные тройки различны.

Гарантируется, что сумма m по всем наборам входных данных не превышает $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите **yes**, если существует кривой цикл, и **no** в противном случае.

Пример

стандартный ввод	стандартный вывод
3	yes
4	yes
1 2 3	no
2 3 4	
3 4 1	
4 1 2	
8	
3 2 5	
6 3 7	
7 3 2	
1 5 6	
5 2 6	
7 4 6	
3 2 6	
3 2 1	
7	
7 4 6	
2 3 7	
7 2 4	
5 4 6	
3 6 7	
5 3 6	
1 4 6	

Задача G. ДИАГОНАЛЬ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

*Так сюда и туда. А теперь диагональ,
диа-го-наль. Диагональ... ДИАГОНАЛЬ...*

— Лосяш

В этот раз вместо легенды будет цитата, а условие будет формальным.

Вам дали таблицу из символов n на n . Строки таблицы можно переставлять произвольным образом. Вы хотите сделать диагональную строку лексикографически минимальной.

Диагональная строка это строка длины n , где i -й символ это символ i -й строки в i -м столбце.

Формат входных данных

Каждый тест содержит несколько наборов входных данных. Первая строка содержит целое число t — количество наборов входных данных ($1 \leq t \leq 300$). Описание наборов входных данных следует далее.

Первая строка содержит целое число n — размер таблицы ($1 \leq n \leq 1000$).

Следующие n строк содержат строки длины n , состоящие из строчных латинских символов — описание таблицы.

Гарантируется, что сумма n по всем наборам не превосходит 1000.

Формат выходных данных

Для каждого набора входных данных выведите лексикографически минимальную диагональ.

Пример

стандартный ввод	стандартный вывод
4	b
1	ca
b	abc
2	abca
ca	
cs	
3	
abc	
arc	
agc	
4	
aaaa	
bcca	
bdca	
bbba	