

Задача А. 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	

Задача В. 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 С. 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 D. Мелодия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

В 2077 году роботы, которые захватили мир, поняли, что человеческая музыка не так уж и хороша, поэтому они начали писать свою.

Для написания музыки у роботов есть специальный музыкальный инструмент, который способен издавать n различных звуков. Каждый из них характеризуется громкостью и высотой. Музыкой называется любая последовательность звуков. Музыка является **красивой**, если любые два подряд идущих звука отличаются либо только громкостью, либо только высотой. Музыка является **скучной**, если громкость или высота каких-то трех подряд идущих звуков равна.

Вы хотите написать **красивую, не скучную** музыку, которая по одному разу содержит все звуки, которые способен издавать ваш музыкальный инструмент.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

В первой строке каждого набора входных данных вводится число n ($1 \leq n \leq 2 \cdot 10^5$) — количество звуков, которые способен издавать музыкальный инструмент.

Далее даны n строк, i -я содержит пару чисел v_i, p_i ($1 \leq v_i, p_i \leq 10^9$) — громкость и высота i -го звука соответственно. Гарантируется, что среди всех n звуков нет равных, то есть для любых $i \neq j$ выполняется хотя бы одно из условий $v_i \neq v_j$, либо же $p_i \neq p_j$.

Сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора данных, если написать такую музыку возможно, то выведите «YES», а в следующей строке выведите n чисел — номера звуков в порядке, образующие красивую нескучную музыку. Иначе выведите «NO».

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Пример

стандартный ввод	стандартный вывод
5	YES
4	2 3 4 1
179 239	NO
179 179	YES
239 179	1
239 239	NO
3	YES
1 1	6 4 3 5 7 2 1
2 1	
3 1	
1	
5 7	
5	
1 1	
1 2	
2 1	
2 2	
99 99	
7	
1 1	
1 3	
2 1	
2 2	
3 1	
3 2	
3 3	

Замечание

В первом наборе входных данных музыка (239, 239) – (239, 179) – (179, 179) – (179, 239) является подходящей, содержит все звуки и все подряд идущие звуки отличаются либо только громкостью, либо же высотой.

Во втором наборе входных данных можно показать, что подходящей музыки с данными звуками не существует.

Задача Е. Варден

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

Замок Вардена представляет собой полный неориентированный граф из N вершин, пронумерованных числами от 1 до N .

Однажды в замок пришли Стив и Алекс и разрушили M рёбер. После этого Варден проснулся в вершине 1, и ему срочно потребовалось добраться до вершины N .

Варден хочет узнать, существует ли путь из вершины 1 в вершину N . Если такой путь существует, его также интересует количество кратчайших путей из вершины 1 в вершину N по модулю 998244353.

Варден не очень хорош в арифметике, поэтому вам придётся ответить на его вопрос))))

Формат входных данных

В первой строке содержатся два целых числа N и M ($2 \leq N \leq 2 \cdot 10^5$, $0 \leq M \leq \min(2 \cdot 10^5, \frac{N(N-1)}{2})$) — количество вершин графа и количество разрушенных рёбер соответственно.

В следующих M строках содержатся по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$) — номера вершин, между которыми было разрушено ребро.

Гарантируется, что все разрушенные рёбра различны.

Формат выходных данных

Если после удаления рёбер путь из вершины 1 в вершину N не существует, выведите -1 .

Иначе выведите количество кратчайших путей из вершины 1 в вершину N по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
6 7 4 3 1 3 2 4 1 6 4 6 5 1 6 2	3
4 6 1 2 1 3 1 4 2 3 2 4 3 4	-1

Задача F. Меньше батареек

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

В 2077 году, когда роботы захватили мир, они решили посоревноваться в следующей игре.

Есть n контрольных точек, причем в i -й из них находится b_i батареек. Изначально Робот находится в 1-й точке без батареек и должен добраться до n -й.

Всего есть m односторонних проходов между точками. i -й из них позволяет переместиться от точки s_i в точку t_i ($s_i < t_i$), но не наоборот. Причем пройти по i -му проходу можно только если у робота есть хотя бы w_i заряженных батареек, иначе он разрядится по пути.

Когда робот оказывается в точке v , он может дополнительно взять любое количество батареек от 0 до b_v включительно. А также он всегда носит все ранее подобранные батарейки, а в каждой контрольной точке он заряжает все ранее подобранные батарейки.

Найдите минимальное количество батареек, которое может быть у робота в конце пути, или сообщите, что добраться от первой точки до последней невозможно.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа n, m ($2 \leq n \leq 2 \cdot 10^5, 0 \leq m \leq 3 \cdot 10^5$) — количество контрольных точек и количество проходов соответственно.

Во второй строке даны n чисел b_i ($0 \leq b_i \leq 10^9$) — количество батареек в i -й контрольной точке.

Следующие m строк содержат по три целых числа s_i, t_i, w_i ($1 \leq s_i < t_i \leq n, 1 \leq w_i \leq 10^9$) — конечные точки прохода и минимальное количество батареек, необходимое для прохода по нему.

Гарантируется, что сумма n не превосходит $2 \cdot 10^5$.

Гарантируется, что сумма m не превосходит $3 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите минимальное количество батареек, которое может оказаться у вас в конце пути, или -1 , если добраться до точки n невозможно.

Пример

стандартный ввод	стандартный вывод
4	1
3 3	4
2 0 0	-1
1 2 1	10
2 3 1	
1 3 2	
5 6	
2 2 5 0 1	
1 2 2	
1 3 1	
1 4 3	
3 5 5	
2 4 4	
4 5 3	
2 0	
1 1	
4 4	
3 10 0 0	
1 2 1	
1 3 3	
2 3 10	
3 4 5	

Замечание

В первом наборе входных данных нужно взять 1 батарейку в стартовой точке, затем перейти в точку 2, а затем в 3.

Во втором наборе входных данных нужно взять 2 батарейки в стартовой точке, затем перейти в точку 2, взять еще 2 батарейки, пройти в точку 4, а затем в точку 5.

В третьем наборе входных данных пути из точки 1 в точку n не существует.

В четвертом наборе входных данных нужно взять 1 батарейку в стартовой точке, затем перейти в точку 2, взять 9 батареек, перейти в точку 3, а затем в точку 4.

Задача G. План эвакуации

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

По прогнозу сейсмологов в Битландии ожидается сильное землетрясение. В стране n городов, пронумерованных от 1 до n , некоторые из которых соединены двусторонними дорогами. Маршрут — это последовательность городов, где каждая пара соседних городов соединены дорогой. Длина маршрута считается как сумма всех длин дорог присутствующих в маршруте. Минимальный маршрут между двумя городами (a, b) — это маршрут с наименьшей длиной, который начинается в городе a и заканчивается в городе b .

Руководство страны главной проблемой считает утечку радиации из атомных электростанций (АЭС) — в этом случае понадобится эвакуация населения. Каждая АЭС расположена в одном из городов и их общее количество равно k , никакой город не содержит более одной АЭС. Руководство хочет составить план эвакуации, который сработает при землетрясении.

Маршрут эвакуации между городами необходимо выбрать так, чтобы он пролегал как можно *дальше* от всех городов, содержащих АЭС. *Опасность* маршрута считается как минимальное расстояние от всех городов на маршруте до какого либо города с АЭС. Более формально, пусть $(a_1, a_2, \dots, a_s)$ это города на маршруте, а $(g_1, g_2, \dots, g_k)$ это города с АЭС, то *опасность* этого маршрута — это минимальное значение среди всех значений $dist(a_i, g_j)$, где $dist(a, b)$ равняется длине минимального маршрута между a и b .

Дано Q пар городов (s_i, t_i) , для которых надо составить план эвакуации с максимально возможной *опасностью*.

Формат входных данных

В первой строке входных данных содержатся целые числа n и m ($2 \leq n \leq 10^5, 1 \leq m \leq 5 \cdot 10^5$) — количество городов и количество дорог в Битландии. Далее в m строках содержатся описания дорог по одной в строке. Каждая дорога задается тремя целыми числами a_i, b_i, w_i ($1 \leq a_i, b_i \leq n, 1 \leq w_i \leq 1000, a_i \neq b_i$) — пара соединенных дорогой городов и длина дороги. Следующая строка содержит одно целое число k ($1 \leq k \leq n$) — количество городов с АЭС. На следующей строке заданы k целых чисел g_i ($1 \leq g_i \leq n$, для всех $1 \leq i \leq k$) — номера городов, в которых расположены АЭС. Следующая строка содержит одно целое число Q ($1 \leq Q \leq 10^5$) — количество пар городов плана эвакуации. Далее в Q строках содержатся (s_i, t_i) ($1 \leq s_i, t_i \leq n, s_i \neq t_i$) — i -я пара городов.

Гарантируется, что никакая дорога не соединяет город с самим собой, между парой городов может быть не более одной дороги и от любого города до любого есть маршрут по существующим дорогам.

Формат выходных данных

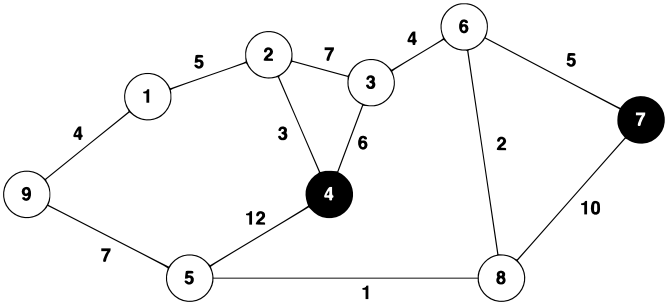
Выведите Q строк.

В i -й строке выведите одно число — максимальная опасность для пары (s_i, t_i) .

Пример

стандартный ввод	стандартный вывод
9 12	5
1 9 4	5
1 2 5	0
2 3 7	7
2 4 3	8
4 3 6	
3 6 4	
8 7 10	
6 7 5	
5 8 1	
9 5 7	
5 4 12	
6 8 2	
2	
4 7	
5	
1 6	
5 3	
4 8	
5 8	
1 5	

Замечание



Задача Н. Блокировка дорожек

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Берендеевы поляны состоят из n домиков, соединенных m двусторонними дорожками. Домики пронумерованы от 1 до n , дорожки от 1 до m . Дорожка номер i соединяет домики с номерами a_i и b_i и имеет длину c_i . Гарантируется, что домик n достижим по дорожкам из домика 1.

Для каждого i от 1 до m определите, различны ли следующие две величины:

- Кратчайшее расстояние от домика 1 до домика n , если все дорожки доступны.
- Кратчайшее расстояние от домика 1 до домика n , если доступны остальные $m - 1$ дорожек, кроме дорожки номер i .

Если домик n достижим из домика 1 в одном случае и не достижим в другом, то величины считаются различными.

Формат входных данных

В первой строке даны два целых числа n и m — количество домиков и дорожек в Берендеевых полянах соответственно ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 2 \cdot 10^5$)

Затем следуют m строк, i -я строка содержит три целых числа a_i, b_i и c_i ($1 \leq a_i < b_i \leq n, 1 \leq c_i \leq 10^9$)

Все пары (a_i, b_i) различны.

Формат выходных данных

Выведите m строк. Строка i должна содержать Yes, если кратчайшее расстояние от домика 1 до домика n , когда все дороги доступны, отличается от кратчайшего расстояния, когда доступны остальные $m - 1$ дорожек, кроме дорожки номер i , иначе — No.

Если домик n достижим по дорожкам в одном случае и недостижим в другом, значения считаются различными.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 5 1 3 10 2 3 6	No Yes No
4 6 2 3 1 2 4 1 3 4 1 1 2 1 1 3 1 1 4 1	No No No No No Yes
2 1 1 2 1	Yes

Замечание

В первом примере, когда все дорожки доступны, расстояние от домика 1 до домика 3 равно 10.

- Когда доступны две дорожки, кроме дорожки 1, кратчайшее расстояние равно 10.
- Когда доступны две дорожки, кроме дорожки 2, кратчайшее расстояние равно 11.

- Когда доступны две дорожки, кроме дорожки 3, кратчайшее расстояние равно 10.

Во втором примере, когда все дорожки доступны, кратчайшее расстояние от домика 1 до домика 4 равно 1. Когда доступны две дорожки, кроме дорожки 6, кратчайшее расстояние равно 2.