

Задача А. Перераспределение камней

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

Как известно, у красавицы и чудовища не все сразу было хорошо. Эта история как раз про это. Как только красавица стала пленницей в замке чудовища, он дал ей первое, но сразу же очень ответственное задание.

Перед красавицей стояло бесконечное количество сундуков, выставленных в линию и пронумерованных целыми числами от $-\infty$ до ∞ . В n сундуках лежали волшебные камни, способные как убивать, так и воскрешать кого угодно, остальные же сундуки были пустые. Задание красавицы состояло в перекладывании камней из сундуков так, чтобы они все в конце концов лежали в n различных сундуках с последовательными номерами. За одно перекладывание красавица могла взять камень из любого сундука и переложить его в любой другой не занятый камнем сундук.

Конечно, красавице захотелось как можно быстрее выполнить ее задание, поэтому она решила минимизировать количество перекладываний. С просьбой найти число этих перекладываний она обратилась к великому волшебнику Мерлину, воззвав к нему о помощи. Однако, даже Мерлин не смог справиться с этой задачей, и ему пришлось проделать долгое путешествие сквозь пространство и время, чтобы попросить помощи у вас. Помогите волшебнику решить задачу красавицы!

Формат входных данных

В первой строке содержится число n — количество сундуков с волшебными камнями ($1 \leq n \leq 10^5$).

Во второй строке содержатся n чисел a_i — номера сундуков с камнями ($-10^9 \leq a_i \leq 10^9$). Гарантируется, что в каждом сундуке лежит не более одного камня.

Формат выходных данных

В единственной строке выведите минимальное количество перекладываний, которое требуется, чтобы разместить все n камней в n различных сундуках с последовательными номерами.

Система оценки

Подзадача	Доп. ограничения	Баллы	Необходимые подзадачи
1	$1 \leq n \leq 1000, -10^4 \leq a_i \leq 10^4$	26	У
2	$1 \leq n \leq 1000$	33	У, 1
3	—	41	У, 1, 2

Пример

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

Замечание

В первом тестовом примере подходит например такой алгоритм действий:

- Переложить камень из сундука -2 в сундук 2
- Переложить камень из сундука 7 в сундук 5

Также можно вторым действием переложить камень из сундука 7 в сундук 0.

Задача В. Площади и фонари

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

В Барселоне 15 века было n площадей. Некоторые площади соединены двусторонней дорогой. Всего существует $n - 1$ таких дорог, причем от каждой площади можно добраться до любой другой по этим дорогам, иначе говоря — площади образуют дерево с n вершинами.

На i -й площади находится r_i фонарей. Каллуму будет легче бороться с тамплиерами, если город будет более освещен. Поэтому он хочет включить на некоторых площадях фонари, причем чтобы i -я площадь была достаточно освещена, на ней должно быть включено не менее l_i фонарей.

Каллум называет площадь *окраинной*, если она соединена ровно одной дорогой с некоторой другой.

Каллум называет площадь *яркой*, если существует способ включить некоторое количество фонарей на каждой из площадей, чтобы все площади были достаточно освещены, и количество горящих фонарей на пути от площади v до всех *окраинных* площадей неравных v , было одинаково.

Количество горящих фонарей на пути — это суммарное количество горящих фонарей на всех площадях этого пути.

Помогите Каллуму определить, какие площади являются яркими.

Формат входных данных

В первой строке находится натуральное число n — количество площадей ($2 \leq n \leq 10^5$).

В следующих $n - 1$ строках находится описание дорог между площадями: в i -й строке два натуральных числа a_i и b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) — номера площадей, которые соединяет данная дорога.

В следующих n строках находится описание фонарей на площадях: в i -й строке два неотрицательных целых числа l_i и r_i ($0 \leq l_i \leq r_i \leq 10^4$) — минимальное и максимальное количество фонарей, которые можно включить на i -й площади.

Гарантируется, что площади образуют дерево.

Формат выходных данных

В единственной строке выведите n чисел: 1 если данная площадь является яркой, и 0 иначе.

Система оценки

Подзадача	Доп. ограничения	Баллы	Необходимые подзадачи
1	$n \leq 15, r_i \leq 1$	21	У
2	$n \leq 2000$	31	У, 1
3	—	48	У, 1, 2

Примеры

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

Задача С. Завод

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

Бэтмен — успешный миллиардер, бизнесмен и супергерой. Для сохранения порядка в городе ему необходимо использовать все чудеса современной техники.

Для создания гаджетов используют самые современные технологии. Завод по производству техники состоит из n конвейеров и m этапов производства. На каждом этапе производства предметы остаются на своем месте либо переходят на один из конвейеров, причем в каждый момент времени на одном конвейере находится ровно один предмет.

Изначально на всех m этапах предметы не меняются местами, то есть после прохождения этапа все предметы остаются на своем месте.

Со временем технологии меняются и необходимо перестраивать завод.

Существуют два типа запросов:

1. a, b, x — Пусть после этапа x предмет с конвейера a попадает на A , а с b на B . Тогда после применения запроса A и B меняются местам, то есть предмет с конвейера a попадает на B , а с b на A .
2. r, x — Вам необходимо узнать на каком конвейере окажется предмет после этапа x , если изначально он находился на конвейере r .

Формат входных данных

В первой строке заданы числа n , m и q — количество конвейеров, этапов и запросов ($1 \leq n, m, q \leq 10^5$).

Каждая из следующих q строк начинается с целого числа t — тип очередного запроса ($0 \leq t \leq 1$). При $t = 0$ запрос первого типа, иначе второго.

Далее в запросах первого типа следует тройка целых чисел a , b и x ($1 \leq a, b \leq n$, $a \neq b$, $1 \leq x \leq m$).

В запросах второго типа следуют целые числа r и x ($1 \leq r \leq n$, $1 \leq x \leq m$).

Формат выходных данных

Для каждого запроса второго типа выведите ответ на него в отдельной строке.

Система оценки

Подзадача	Доп. ограничения	Баллы	Необходимые подзадачи
1	$n, m, q \leq 100$	12	У
2	$m, q \leq 5000$	14	У, 1
3	$n = 2$	25	—
4	$n, m, q \leq 25000$	30	У, 1, 2
5	—	19	У, 1, 2, 3, 4

Примеры

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

Задача D. Algotcode.Еда

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	11.999 секунд
Ограничение по памяти:	999 мегабайт

Даня и Филипп решили запустить новый раздел сайта: «Algotcode.Еда». Курьерами было решено назначить школьников из кружка, особо шумящих во время занятий. Всего сервис будет обслуживать n домов, соединенных m дорогами. Дома пронумерованы от 1 до n , а дороги — от 1 до m . Дорога с номером i соединяет дома a_i и b_i в обоих направлениях, при этом перемещение по ней занимает c_i минут. По дорогам можно добраться от любого дома до любого другого.

К сожалению, из-за очень большого размера рюкзака, курьеры не смогут совершать U-образные повороты. А именно, курьер не может посреди дороги развернуться и пойти в обратном направлении. Более того, он не может прибыть в дом, а затем пойти по той же дороге, откуда пришел. (Да, рюкзак настолько большой.)

Каждый день курьер разносит еду в соответствии с **планом доставки**. План доставки — последовательность из k домов x_1, x_2, \dots, x_k . Курьер начинает в доме x_1 , и далее посещает дома в соответствующем порядке, завершая в доме x_k . Ему разрешается проходить мимо других домов между посещением домов из плана. Возможно, что в какой-то дом ему придется доставить еду несколько раз, но при этом гарантируется, что $x_i \neq x_{i+1}$ для всех i ($1 \leq i \leq k-1$). Обратите внимание, что план доставки может быть таким, что курьер не может его выполнить. (За что, конечно, будет оштрафован.)

В самом начале курьер выполняет исходный план доставки x_1, x_2, \dots, x_k . Утром i -го дня ($1 \leq i \leq t$) он изменит p_i -е значение плана доставки на q_k (то есть x_{p_i} становится равным q_i , и затем разносит еду в соответствии с новым планом доставки. Гарантируется, что после каждого изменения $x_j \neq x_{j+1}$ для всех j ($1 \leq j \leq k-1$).

Для каждого плана доставки для этих t дней курьер хочет знать, можно ли его выполнить, и, если можно, то за какое минимальное время можно разнести всю еду в соответствии с планом.

Формат входных данных

Первая строка содержит четыре целых числа n, m, t и k ($2 \leq n \leq 2000, n-1 \leq m \leq 2000, 1 \leq t \leq 10^5, 2 \leq k \leq 10^5$).

Следующие m строк описывают дороги, i -я из них содержит три целых числа a_i, b_i и c_i ($1 \leq a_i < b_i \leq n, (a_i, b_i) \neq (a_j, b_j)$ для $i \neq j, 1 \leq c_i \leq 10^9$).

Следующие k строк описывают изначальный план доставки, i -я из них содержит целое число x_i ($1 \leq x_i \leq n$).

Следующие t строк описывают изменения плана доставки, i -я из них содержит два целых числа p_i и q_i ($1 \leq p_i \leq k, 1 \leq q_i \leq n$).

Гарантируется, что из любого дома можно добраться до любого другого, используя одну или более дорог. Гарантируется, что $x_i \neq x_{i+1}$ для всех i ($1 \leq i \leq k-1$) как изначально, так и после каждого изменения.

Формат выходных данных

Выведите k строк, i -я из которых содержит -1 , если такой план доставки не осуществим, и минимальное время в минутах на его осуществление в противном случае.

Система оценки

Подзадача	Доп. ограничения	Баллы	Необходимые подзадачи
1	$n, m, k, c_i \leq 10, t = 1$	12	—
2	$n, m \leq 500, t = 1$	35	1
3	$t = 1$	15	2, 3
4	—	38	1, 2, 3

Примеры

стандартный ввод	стандартный вывод
3 3 1 3 1 2 1 2 3 1 1 3 1 1 2 3 3 1	3
4 4 4 3 1 2 1 2 3 1 1 3 1 1 4 1 4 1 3 3 4 1 2 3 2 2 4	5 2 3 -1
5 6 1 5 1 2 8 1 3 8 1 4 8 2 5 2 3 4 6 4 5 6 2 5 1 5 3 5 2	38