

Задача А. Праздник к нам приходит

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

В уездном городе Т все заняты подготовкой к новому году. Таня вот, например, готовится вручать подарки в новогоднюю ночь (в городе Т именно Таня исполняет роль Деда Мороза).

В городе Т n домов, некоторые из которых соединены улицами, причем так, что между любыми двумя домами есть ровно один путь.

Таня уже заготовила m мешков с подарками (в этом году она дарит серые футболки с желтым единорогом), но столкнулась с проблемой. Ей ведь самой тоже нужно найти дом в городе Т для празднования Нового Года. Раздачей подарков же, как обычно, будут заниматься олениа. Одному олененку можно дать ровно один мешок и отправить его в путь. При этом олениа не ходят по одной и той же улице дважды. i -й мешок предназначен для жителей всех домов на пути от a_i до b_i . Поэтому считается, что олененку можно дать i -й мешок с подарками, если он сможет выйти из места празднования Тани и пройти через все дома на пути от a_i до b_i , при этом не проходя через одну и ту же улицу дважды.

Помогите Тани найти дом для празднования так, чтобы она смогла отправить как можно больше мешков с подарками. В данной задаче можно считать, что количество олениа не ограничено.

Формат входных данных

В первой строке задано число n — количество домов в городе Т ($2 \leq n \leq 2 \cdot 10^5$).

В следующих $n - 1$ строках описаны улицы. Улица задаётся числами x_i и y_i — номерами домов, которые она соединяет ($1 \leq x_i, y_i \leq n, x_i \neq y_i$). Гарантируется, что между любыми двумя домами существует единственный путь.

В следующей строке задано число m — количество мешков у Тани ($1 \leq m \leq 2 \cdot 10^5$).

В следующих m строках описаны мешки с футболками. В i -й из них заданы числа a_i и b_i — начало и конец i -го пути ($1 \leq a_i, b_i \leq n, a_i \neq b_i$). Пути могут пересекаться и совпадать.

Дома нумеруются с единицы.

Формат выходных данных

Выведите одно число — максимальное число мешков с подарками, которые Таня сможет раздать, если выберет оптимальный дом для празднования.

Пример

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

Задача В. Вирусы и антивирусы

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

Антивирусная IT-компания имеет официальную иерархическую структуру управления. В ней есть босс — единственный сотрудник, над которым нет начальника. Каждый из остальных сотрудников подчинён ровно одному сотруднику — своему начальнику. Начальник может иметь нескольких подчинённых и отдавать или передавать приказы любому из них. Приказы могут передаваться от одного сотрудника другому только по цепочке, каждый раз от начальника к его подчинённому. Сотрудник А главнее сотрудника Б в этой иерархии, если А может отдать или передать приказ сотруднику Б непосредственно, или через цепочку подчинённых. Босс главнее любого сотрудника.

Оказалось, что все сотрудники объединены ещё в одну организованную подобным образом тайную иерархическую структуру, производящую компьютерные вирусы. В тайной структуре может быть другой босс, а у сотрудников — другие начальники.

Будем называть пару сотрудников А и Б устойчивой, если А главнее Б и в основной, и в тайной иерархических структурах.

Требуется написать программу, определяющую количество устойчивых пар в компании.

Формат входных данных

В первой строке задано число N — количество сотрудников компании ($1 \leq N \leq 100\,000$).

Во второй строке — N целых чисел a_i , где $a_i = 0$, если в официальной иерархии сотрудник с номером i является боссом, в противном случае a_i равно номеру непосредственного начальника сотрудника номер i .

В третьей строке — N целых чисел b_i , где $b_i = 0$, если в тайной иерархии сотрудник с номером i является боссом, в противном случае b_i равно номеру непосредственного начальника сотрудника номер i .

Нумерация сотрудников ведется с единицы в том порядке, в каком они упомянуты во входном файле.

Формат выходных данных

Выходной файл должен содержать единственное число — количество устойчивых пар.

Примеры

стандартный ввод	стандартный вывод
3 0 3 1 0 1 1	2
5 2 0 1 3 4 3 1 0 2 4	7

Задача С. Очередные запросы на дереве

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

Вам дано дерево, состоящее из n вершин. Каждое ребро дерева имеет длину, которая является некоторым натуральным числом. Вам нужно ответить на q запросов, каждый задается парой вершин a и b и числом x . Ответом на запрос является количество ребер на пути от a до b , длина которых $\leq x$.

Формат входных данных

В первой строке находится два целых числа n и q ($2 \leq n, q \leq 200\,000$), разделенных пробелом. В следующих $n - 1$ строках находится по три целых числа s, f, l ($1 \leq s, f \leq n, 1 \leq l \leq 10^6$), разделенных пробелами, обозначающих ребро (s, f) длины l . В следующих q строках находится по три целых числа a, b, x ($1 \leq a, b \leq n, 1 \leq x \leq 10^6, a \neq b$), разделенных пробелами, обозначающих запрос a, b, x .

Формат выходных данных

Выведите q строк, в i -й строке ответ на i -й запрос.

Пример

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

Задача D. Антивещество

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	128 мегабайт

Компания тестирует технологию получения антивещества, используемого в качестве топлива в межпланетном звездолёте. Антивещество получается в результате специальных экспериментов в реакторе.

Известно n типов экспериментов, приводящих к получению антивещества. В результате проведения эксперимента i -го типа в выходной контейнер реактора добавляется от l_i до r_i граммов антивещества. Из соображений безопасности запрещается накапливать в контейнере более a граммов антивещества.

Затраты на проведение эксперимента i -го типа составляют c_i , а стоимость одного грамма полученного антивещества составляет 10^9 .

Если после проведения экспериментов в контейнере образовалось t граммов антивещества, а суммарные затраты на проведение экспериментов в реакторе составили s , то прибыль определяется по формуле $(t \cdot 10^9 - s)$. Компании необходимо разработать стратегию проведения экспериментов, позволяющую максимизировать прибыль, которую можно гарантированно получить.

В зависимости от результатов предыдущих экспериментов стратегия определяет, эксперимент какого типа следует провести, или решает прекратить дальнейшее выполнение экспериментов.

Стратегия позволяет гарантированно получить прибыль x , если при любых результатах проведения экспериментов: во-первых, в контейнере реактора оказывается не более a граммов антивещества, во-вторых, прибыль составит не менее x . Например, пусть возможен только один тип эксперимента, порождающий от 4 до 6 граммов антивещества, затраты на его проведение равны 10, а вместимость контейнера составляет 17 граммов. Тогда после двукратного проведения эксперимента в контейнере может оказаться от 8 до 12 граммов антивещества. Если получилось 12 граммов, то больше проводить эксперимент нельзя, так как в случае получения 6 граммов антивещества контейнер может переполниться. В остальных случаях можно провести эксперимент в третий раз и получить от 12 до 17 граммов антивещества. В худшем случае придётся провести эксперимент трижды, затратив в сумме 30, прибыль составит $(12 \cdot 10^9 - 30) = 11999999970$.

Требуется написать программу, которая определяет максимальную прибыль x , которую гарантированно можно получить.

Формат входных данных

Первая строка входных данных содержит два целых числа: n — количество типов экспериментов и a — максимально допустимое количество антивещества в контейнере ($1 \leq n \leq 100, 1 \leq a \leq 2000000$).

Следующие n строк содержат по три целых числа l_i, r_i, c_i — минимальное и максимальное количество антивещества, получаемое в результате эксперимента типа i , и затраты на эксперимент этого типа, соответственно ($1 \leq l_i \leq r_i \leq a, 1 \leq c_i \leq 100$).

Формат выходных данных

Выходные данные должны содержать одно целое число — максимальную прибыль x , которую гарантированно можно получить.

Примеры

стандартный ввод	стандартный вывод
1 17 4 6 10	11999999970
2 11 2 2 100 3 5 5	9999999890
1 1 1 1 1	999999999

Задача E. Сложная задача

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	768 мегабайт

У Егора есть большой лист клетчатой бумаги и не менее большое домашнее задание по математике. Егор, как недобросовестный ученик, планирует списать домашнюю работу с сервиса Титан Бета.

Решение задачи Егора в системе Титан Бета состоит из n формул-шагов, каждую из которых он должен перенести на лист. Разумеется, формулы в листе должны идти по порядку — слева направо. Если в какой-то момент написать следующую формулу нельзя — её обязательно надо перенести на следующую строку. Но также разрешено делать переносы и в произвольном месте решения, на усмотрение ученика, однако формула должна быть написана целиком на одной строке.

Поскольку в формулах могут возникать дробные выражения, высоты формул бывают разными. Егор относится к формулам, как к бессвязному набору символов, который влезает в какой-то клетчатый прямоугольник; а именно, i -я формула занимает прямоугольник w_i клеток в ширину и h_i клеток в высоту.

Чтобы строки читались аккуратно, Егор выравнивает все формулы, записанные в одной строке, по нижней границе строки. Таким образом, высота строки равна высоте самой высокой формуле строки. Ширина же строки равна суммарной ширине формул в строке. Суммарной высотой решения назовем суммарную высоту всех строк.

Ширина листа бумаги составляет ровно m клеток. Егора заинтересовало, какую наименьшую высоту может иметь его решение? Поскольку ему надо делать домашнюю работу, то ответ на этот вопрос предстоит найти вам.

Формат входных данных

В данной задаче ввод устроен необычным образом — для тестов последней группы часть входного файла вам придется генерировать самостоятельно.

В первой строке вводятся три числа n, m, k ($1 \leq n \leq 10^7$, $1 \leq m \leq 10^{11}$, $1 \leq k \leq \min(500\,000, n)$) — общее количество формул, ширина листа, количество формул, высота и ширина для которых будут введены далее.

В следующих k строках вводится по три целых числа i_j, w_{i_j}, h_{i_j} ($1 \leq i_j \leq n$, $1 \leq w_{i_j}, h_{i_j} \leq m$). Первое число — номер одной из формул Егора. Следующие два числа — ширина и высота этой формулы соответственно.

Гарантируется, что индексы i_j возрастают с каждой следующей строкой.

В последней строке вводятся 4 числа a, b, c, d ($1 \leq a, c \leq \min(10\,000, m)$, $0 \leq b, d \leq m$) — вспомогательные параметры, которые отвечают за генерацию остальных входных данных.

Для тех формул, номера которых не были перечислены во входных данных, высоту и ширину придётся генерировать самостоятельно. А именно, для любого i , не перечисленного во входных данных, $w_i = ((w_{i-1} \cdot a + b) \bmod m) + 1$, $h_i = ((h_{i-1} \cdot c + d) \bmod m) + 1$. Гарантируется, что $i = 1$ присутствует во входных данных (соответственно, размеры всех формул определяются однозначно).

Формат выходных данных

Выведите одно число — минимальную высоту листа шириной m , в который получится уместить все формулы.

Примеры

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

Замечание

В первом примере из условия оптимальным вариантом размещения будет тот, при котором первая формула находится на отдельной строке, а вторая и третья — на следующей. Таким образом, суммарная высота получается равной $2 + 3 = 5$ клеткам.

Во втором примере из условия прямоугольники имеют вид $[(2, 3), (1, 2), (3, 1), (2, 3)]$. Третий прямоугольник обязательно займет целую строку. Четвертая формула, таким образом, займет свою отдельную строчку, а первая и вторая формулы сгруппируются. Получившиеся строки будут вместе занимать $3 + 1 + 3 = 7$ клеток в высоту.

Задача F. Плитка для ванной

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 512 мегабайт

У Кости очень много дел — ремонт в самом разгаре! Надо клеить обои, собирать мебель и постоянно вывозить мусор.

Сегодня Костя хочет купить плитку для ванной. Он пришел в магазин и оказался перед большим квадратным стендом с плиткой. Стенд представляет из себя квадрат из $n \times n$ клеток, каждая клетка которого — маленький кусочек плитки цвета $c_{i,j}$. Магазин продает кусочки плитки пакетами — а именно, купить можно только подквадрат исходного квадрата.

Подквадратом называется любой квадратный фрагмент стенда, то есть любое множество $S(i_0, j_0, k) = \{c_{i,j} \mid i_0 \leq i < i_0 + k, j_0 \leq j < j_0 + k\}$ при $1 \leq i_0, j_0 \leq n - k + 1$.

Костя еще не знает, сколько кусочков плитки он хочет купить, и, соответственно, рассматривает подквадраты всех возможных размеров. При этом он точно не хочет слишком разноцветную плитку в ванной, что позволяет ему сузить выбор. Помогите Косте для каждого $k \leq n$ определить количество различных подквадратов размера $k \times k$, в которых не более q различных цветов плитки. Подквадраты считаются различными, если их расположение на стенде не совпадает.

Формат входных данных

В первой строке вводятся два целых положительных числа n, q ($1 \leq n \leq 1500, 1 \leq q \leq 10$) — размер стенда с плитками и ограничение на количество различных цветов в пакете.

В следующих n строках вводятся по n целых положительных чисел $c_{i,j}$ ($1 \leq c_{i,j} \leq n^2$) — j -е число в i -й строке соответствует цвету плитки в клетке (i, j) .

Формат выходных данных

Для каждого k от 1 до n выведите в отдельной строке по одному целому числу — количество подквадратов размера $k \times k$, которые интересны Косте.

Примеры

стандартный ввод	стандартный вывод
3 4 1 2 3 4 5 6 7 8 9	9 4 0
4 8 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7	16 9 4 0

Замечание

В первом примере все цвета квадратиков плитки различные. Поскольку Костя не хочет, чтобы в купленном квадрате было больше 4 цветов, он может купить себе любой подквадрат размера 1×1 или 2×2 , но при этом не сможет купить квадрат размера 3×3 .

Во втором примере есть повторяющиеся цвета. А именно, за счет ограничения $q = 8$ Костя может купить любой подквадрат 1×1 и 2×2 , а также любой подквадрат 3×3 , ведь в каждом таком подквадрате всего 7 цветов. Весь стенд размера 4×4 Костя купить не сможет, потому что там будет 9 цветов.

Система оценки