

Задача А. Система непересекающихся множеств

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

Реализуйте систему непересекающихся множеств. Кроме того, вам нужно будет найти для множества минимум, максимум и размер.
Изначально каждый элемент находится в своем собственном множестве.

Формат входных данных

Первая строка ввода содержит целое число n – количество элементов в наборе, m – количество операций ($1 \leq n, m \leq 300000$). Каждая из следующих m строк содержит одну операцию. Существует два типа операций:

- `union x y`: объединить множества, содержащие элементы x и y .
- `get x`: найти множество, содержащее элемент x .

Формат выходных данных

Для каждой операции `get x` выведите минимум, максимум и размер множества, содержащего элемент x .

Пример

стандартный ввод	стандартный вывод
5 11	3 3 1
union 1 2	1 2 2
get 3	1 3 3
get 2	5 5 1
union 2 3	4 5 2
get 2	1 5 5
union 1 3	
get 5	
union 4 5	
get 5	
union 4 1	
get 5	

Задача В. Остовное дерево

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

Требуется найти в связном графе остовное дерево минимального веса.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m – количество вершин и ребер графа соответственно. Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается тремя натуральными числами b_i , e_i и w_i – номера концов ребра и его вес соответственно ($1 \leq b_i, e_i \leq n, 0 \leq w_i \leq 100000, n \leq 200000, m \leq 200000$).

Граф является связным.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число – вес минимального остовного дерева.

Пример

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

Задача С. Печатная схема

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

Печатной схемой называется плоская поверхность, содержащая узлы и перемычки, соединяющие пары узлов. Мы будем рассматривать печатные схемы специального вида, где все узлы расположены в узлах прямоугольной сетки, а перемычки (вертикальные или горизонтальные) соединяют пары соседних узлов. Печатная схема называется связной, если любые два узла соединены друг с другом последовательностью перемычек. На вход дается печатная схема, где некоторые соседние узлы уже соединены перемычками. К ней необходимо добавить некоторое количество перемычек таким образом, чтобы схема стала связной. Стоимость вертикальной перемычки – 1, а горизонтальной – 2.

Ваша программа должна по начальной печатной схеме определить количество добавляемых перемычек, минимальную стоимость такого добавления, а также вывести сами добавляемые перемычки.

Формат входных данных

Первая строка входного файла содержит два натуральных числа N и M – количество строк и количество столбцов соответственно ($1 \leq N, M \leq 100$). Каждый узел определяется своими координатами, нумерация начинается с верхнего левого угла (координаты $(1, 1)$). Далее дается описание решетки в виде N строк по M чисел. Каждое число обозначает связь узла (i, j) с узлами $(i + 1, j)$ и $(i, j + 1)$ в следующем формате:

- 0 – узел (i, j) не соединен ни с одним из узлов $(i + 1, j)$ и $(i, j + 1)$
- 1 – узел (i, j) соединен только с узлом $(i + 1, j)$
- 2 – узел (i, j) соединен только с узлом $(i, j + 1)$
- 3 – узел (i, j) соединен и с узлом $(i + 1, j)$, и с узлом $(i, j + 1)$.

Формат выходных данных

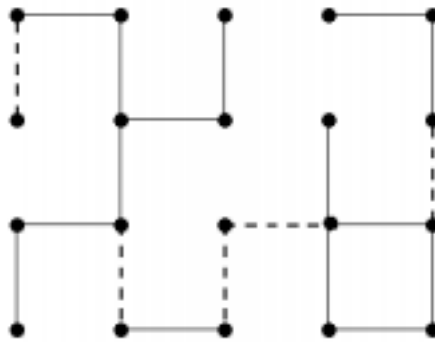
Первая строка выходного файла должна содержать два числа K и V – количество добавленных перемычек и стоимость добавления соответственно. Каждая из следующих K строк должна содержать описание добавленной перемычки в формате i, j, d , где (i, j) – координаты начального узла, а d может принимать значения 1 или 2. $d = 1$ обозначает, что соединены узлы (i, j) и $(i + 1, j)$, $d = 2$ – узлы (i, j) и $(i, j + 1)$. Если решений несколько – выведите любое из них.

Пример

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

Замечание

Пояснение к первому примеру:



Задача D. Безумные монстры

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

Монокарп играет в компьютерную игру (опять). Угадайте, что он делает? Правильно, убивает монстров.

В ряд стоят n монстров, пронумерованных от 1 до n . У i -го монстра есть два параметра: значение атаки, равное a_i , и значение защиты, равное d_i . Чтобы убить этих монстров, Монокарп наложил на них заклинание безумия, так что они атакуют друг друга, а не персонажа Монокарпа.

Бой состоит из n раундов. В каждом раунде происходит следующее:

- сначала каждый живой монстр i наносит a_i урона ближайшему живому монстру слева (если он существует) и ближайшему живому монстру справа (если он существует);
- затем каждый живой монстр j , который получил больше d_j урона в течение этого раунда, умирает. Т.е. j -й монстр умирает тогда и только тогда, когда его значение защиты d_j строго меньше суммарного урона, который он получил в этом раунде.

Для каждого раунда вычислите количество монстров, которые умрут во время этого раунда.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 3 \cdot 10^5$). Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$). Третья строка содержит n целых чисел d_1, d_2, \dots, d_n ($1 \leq d_i \leq 10^9$).

Формат выходных данных

Выведите n целых чисел. i -е целое число должно быть равно количеству монстров, которые умрут в i -м раунде.

Примеры

стандартный ввод	стандартный вывод
5 3 4 7 5 10 4 9 1 18 1	3 1 0 0 0
2 2 1 1 3	0 0
4 1 1 2 4 3 3 4 2	1 1 1 0

Задача Е. Кратчайший путь

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

Дан неориентированный связный взвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Формат входных данных

В первой строке входного файла два числа: n и m ($2 \leq n \leq 30000, 1 \leq m \leq 400000$), где n – количество вершин графа, а m – количество ребер.

Следующие m строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра – неотрицательное целое число, не превосходящее 10^4 .

Формат выходных данных

Выведите n чисел – для каждой вершины кратчайшее расстояние до нее.

Пример

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

Задача F. Транспортировка

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

К очередной Летней компьютерной школе было решено подготовить кружки как для школьников, так и для всех преподавателей.

Имея привычку делать важные дела в самый последний момент, дизайнер закончил работу над макетом за два дня до начала школы. Ещё день уйдёт у завода-изготовителя на то, чтобы изготовить кружки и нанести на них изображение. На то, чтобы довезти кружки от завода-изготовителя до ЛКШ, остаётся всего 24 часа.

Заказ на десять миллионов экземпляров кружек (а именно столько заказали организаторы), конечно же, за один рейс не увезти. Однако, за первый рейс хочется привезти максимальное количество кружек. Для перевозки был заказан один большегрузный автомобиль. Но есть один нюанс: на некоторых дорогах установлено ограничение на вес автомобиля. Поэтому если автомобиль нагрузить кружками под завязку, то, возможно, не удастся воспользоваться самым коротким маршрутом, и придётся ехать в объезд. Может случиться даже так, что из-за этого грузовик не успеет доехать до лагеря вовремя, а этого допустить никак нельзя.

Итак, сколько же кружек можно погрузить в автомобиль, чтобы успеть привезти этот ценный груз вовремя, и не нарушая правил дорожного движения?

Формат входных данных

В первой строке входного файла находятся числа N и M ($1 \leq N \leq 500, 1 \leq M \leq 100000$) – количество узловых пунктов дорожной схемы и количество дорог, соответственно. В следующих M строках находится информация о дорогах. Каждая дорога описывается в отдельной строке следующим образом. Сначала указаны номера узловых пунктов, которые соединяются данной дорогой, потом время, которое тратится на проезд по этой дороге, и, наконец, максимальный вес автомобиля, которому разрешено ехать по этой дороге. Известно, что все дороги соединяют различные пункты, причем для каждой пары пунктов есть не более одной дороги, непосредственно их соединяющей. По дороге можно проехать в обоих направлениях. Все числа разделены одним или несколькими пробелами.

Узловые пункты нумеруются числами от 1 до N . При этом завод по производству кружек имеет номер 1, а ЛКШ – номер N . Время проезда по дороге задано в минутах и не превосходит 1440 (24 часа). Ограничение на массу задано в граммах и не превосходит одного миллиарда. Кроме того, известно, что одна кружка весит 100 грамм, а пустой грузовик – 3 тонны.

Формат выходных данных

В выходной файл выведите одно число – максимальное количество кружек, которое можно привезти за первый рейс, потратив не более 24 часов.

Пример

стандартный ввод	стандартный вывод
3 3 1 2 10 3000220 2 3 20 3000201 1 3 1 3000099	2

Задача G. В поисках утраченного кефира

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

Школьник Вася хочет найти запасы спрятанного кефира. По легенде, кефир находится в домиках a , b или c . Вася хочет проверить каждый из этих трёх домиков, потратив на это минимальное количество времени.

Местность, в которой находится Вася, представляет собой n домиков, пронумерованных числами от 1 до n . Некоторые из домиков соединены дорогами, по которым можно ходить в обе стороны. Время прохождения i -й дороги составляет w_i секунд. Путём в графе называется непустая последовательность вершин, такая, что все соседние вершины соединены дорогой. Требуется помочь Васе найти путь, содержащий вершины a , b , c , такой, что суммарное время прохождения всех дорог на пути минимально. При этом, если мы прошли по какой-то дороге дважды (или более), то и время её прохождения следует учитывать соответствующее количество раз. Начинать свой путь Вася может из любой вершины.

Гарантируется, что a , b , c — попарно различные домики.

Формат входных данных

В первой строке ввода записаны два числа n и m ($3 \leq n \leq 100000, 0 \leq m \leq 200000$) — количество домиков в ЛКШ и дорог между ними соответственно.

Следующие m строк содержат описания дорог, по одному в строке. Каждая из дорог задаётся тройкой чисел u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^9$) — номерами соединённых домиков и временем, затрачиваемым на прохождение данной дороги. По каждой дороге разрешено ходить в обе стороны. Гарантируется, что любая пара домиков соединена не более чем одной дорогой. Также гарантируется, что нет дороги, соединяющей домик с самим собой.

В последней строке записаны три попарно различных числа a, b, c ($1 \leq a, b, c \leq n$).

Формат выходных данных

Выведите одно целое число — минимальное возможное время, которое нужно затратить на прохождение пути, содержащего домики a, b и c . Если пути, содержащего все три домика, не существует, то выведите -1.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 3 2 3 1 3 4 7 2 4 10 1 3 4	11
4 2 1 2 10 2 3 5 1 2 4	-1