

Задача А. Проверка маркеров

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

Через пять минут Александр Маркович уже должен начать лекцию, а он еще только входит на ступеньки крыльца университета! Это вполне обычная ситуация, и все было бы не так плохо, если бы лекция не была в той самой огромной аудитории, где все время творится бардак, ведь в ней всегда валяется огромное количество закончившихся маркеров. А перед началом лекции Александру Марковичу нужно найти хотя бы два маркера различных цветов, которые еще не закончились.

Всего в университете пользуются маркерами n цветов, и в этой аудитории все они свалены в одну кучу. Мы знаем, что среди маркеров i -го цвета есть a_i закончившихся и b_i хороших, которыми еще можно писать. По внешнему виду отличить закончившийся маркер от хорошего нельзя. Чтобы найти два хороших маркера разных цветов, Александр Маркович будет повторять следующую процедуру:

1. он возьмет из кучи два маркера различных цветов;
2. затем он одновременно проверит, пишет ли каждый из маркеров;
3. если оба маркера пишут, то Александр Маркович возьмет их и начнет читать лекцию;
4. если же хотя бы один из маркеров не пишет, он выкинет оба маркера в мусорное ведро и вернется к пункту 1.

Александр Маркович выбирает пару маркеров произвольным образом. Может ли случиться такое, что он не сможет найти два хороших маркера различных цветов, то есть при очередном исполнении пункта 1 в куче не будет двух маркеров разных цветов?

Вам необходимо решить задачу для t наборов входных

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 2 \cdot 10^5$) — количество наборов входных данных.

Каждый из наборов входных данных описывается в трех строках. Первая строка содержит одно целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество различных цветов маркеров.

Вторая строка каждого набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) — количество закончившихся маркеров каждого из цветов.

Третья строка каждого набора входных данных содержит n целых чисел b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$) — количество хороших маркеров каждого из цветов.

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

Формат выходных данных

Для каждого набора входных данных выведите на отдельной строке «YES», если может случиться такое, что Александр Маркович не найдет два хороших маркера разных цветов, и «NO» иначе.

Пример

стандартный ввод	стандартный вывод
3	YES
3	NO
1 2 1	YES
2 1 1	
2	
1 1	
2 2	
4	
1 1 1 1	
2 1 2 1	

Задача В. Совершенное паросочетание на дереве

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

Дано дерево T на N вершинах. Вершины пронумерованы числами от 1 до N , и i -е ребро ($1 \leq i \leq N - 1$) соединяет вершины u_i и v_i в обоих направлениях.

Построим полный граф G на N вершинах следующим образом: вес $w(x, y)$ ребра между вершинами x и y в G равен длине кратчайшего пути между x и y в дереве T .

Найдите максимальное паросочетание максимального веса в G . То есть, найдите множество из $\lfloor N/2 \rfloor$ пар вершин $M = \{(x_1, y_1), (x_2, y_2), \dots, (x_{\lfloor N/2 \rfloor}, y_{\lfloor N/2 \rfloor})\}$, такое что каждая вершина $1, 2, \dots, N$ присутствует в M не более одного раза, и $\sum_{i=1}^{\lfloor N/2 \rfloor} w(x_i, y_i)$ максимальна.

Формат входных данных

В первой строке дано число N — количество вершин в дереве T ($1 \leq N \leq 2 \cdot 10^5$)

В следующих $n - 1$ строках дано описание рёбер дерева: i -я строка содержит два числа u_i, v_i — концы i -го ребра ($1 \leq u_i < v_i \leq N$)

Гарантируется, что данные рёбра образуют дерево на N вершинах.

Формат выходных данных

Пусть вы нашли искомое паросочетание $\{(x_1, y_1), (x_2, y_2), \dots, (x_{\lfloor N/2 \rfloor}, y_{\lfloor N/2 \rfloor})\}$, где i -я пара — это i -е ребро из паросочетания. В качестве ответа выведите $\lfloor N/2 \rfloor$ строк, i -я из которых содержит x_i и y_i через пробел. Если существует несколько оптимальных паросочетаний, выведите любое из них.

Примеры

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

Замечание

В первом примере входных данных, например, паросочетание $\{(2, 3), (1, 4)\}$ тоже является оптимальным.

Задача С. Королевские вопросы

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

В одном средневековом королевстве бушует экономический кризис. Надои молока падают, экономические показатели каждый день пробивают новое дно, деньги из казны пропадают в неизвестном направлении. Чтобы исправить положение, король Карл Солнцеликий принял решение женить своих n сыновей-принцев на невестах s , по возможности, как можно большим приданым.

В поисках кандидаток король отправил гонцов в соседние королевства, в результате чего ко двору прибыли делегации с m незамужними принцессами. Принимая гостей, Карл узнал, что приданое i -й принцессы составляет w_i золотых монет.

Хотя дело и происходит в средневековье, в обществе уже широко распространены прогрессивные представления, в соответствии с которыми никто не может заставить принцессу выйти замуж за принца, который ей не по нраву. Поэтому каждой принцессе была предоставлена возможность выбрать двух принцев, за каждого из которых она готова выйти замуж. Принцам повезло меньше, они в вопросе выбора невесты будут повиноваться воле своего отца.

Зная размер приданого и предпочтения каждой принцессы, Карл хочет сыграть свадьбу таким образом, чтобы суммарное приданое невест всех его сыновей было как можно больше. При этом женить всех принцев или выдавать всех принцесс замуж не обязательно. Каждый принц может жениться не более чем на одной принцессе, и наоборот каждая принцесса может быть выдана не более чем за одного принца.

Помогите королю организовать женитьбу своих сыновей наиболее выгодным для казны способом.

Формат входных данных

В первой строке находятся два целых числа n, m ($2 \leq n \leq 200\,000, 1 \leq m \leq 200\,000$) — количество принцев и количество принцесс соответственно.

В каждой из последующих m строк находится по три целых числа a_i, b_i, w_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq w_i \leq 10\,000$) — номера принцев, за которых готова выйти замуж i -я принцесса и размер её приданого.

Формат выходных данных

Выведите единственное целое число — максимальное количество золотых монет, которое может получить король, сыграв нужные свадьбы.

Примеры

стандартный ввод	стандартный вывод
2 3 1 2 5 1 2 1 2 1 10	15
3 2 1 2 10 3 2 20	30

Задача D. Вышивка жемчугом

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

Со стародавних времён в поморских деревнях рукодельницы вышивали жемчугом на прямоугольных полотенцах, состоящих из одинаковых клеток. Вышивка начиналась с пришивания жемчужины к полотенцу в центре одной из клеток. Чтобы пришить новую жемчужину, рукодельница делала стежок из клетки, уже содержащей жемчужину, в соседнюю с ней по горизонтали или вертикали свободную клетку. Новая жемчужина пришивалась в центре клетки на конце стежка. Этот процесс повторялся, пока не заканчивались жемчужины.

Одно из таких праздничных полотенеч находится в музее. К сожалению, некоторые части узора были утеряны, но описание полотенца сохранилось. Дирекция музея планирует восстановить один из прямоугольных фрагментов полотенца, но не ещё не решила какой именно. Затраты на восстановление фрагмента зависят от количества связанных частей узора, попавших на этот фрагмент. Часть узора считается связанной, если от любой её жемчужины можно по стежкам перейти к любой другой её жемчужине, не выходя за границы фрагмента. Дирекция всегда относит любые две жемчужины, между которыми можно перейти по стежкам, к одной и той же связанной части узора.

Требуется написать программу, вычисляющую количество связанных частей узора для каждого из заданных фрагментов.

Формат входных данных

Первая строка входных данных содержит два целых числа a и b — размеры полотенца в клетках по горизонтали и вертикали.

Вторая строка содержит два числа n и q — количество жемчужин в узоре и количество фрагментов соответственно.

Следующие $(n-1)$ строк содержат описания стежков. Каждый стежок имеет один из следующих видов:

- $h\ x\ y$ означает, что клетки с координатами (x, y) и $(x+1, y)$ содержат жемчужины, соединённые горизонтальным стежком ($1 \leq x \leq a-1$; $1 \leq y \leq b$);
- $v\ x\ y$ означает, что клетки с координатами (x, y) и $(x, y+1)$ содержат жемчужины, соединённые вертикальным стежком ($1 \leq x \leq a$; $1 \leq y \leq b-1$).

Так как неизвестно в каком порядке рукодельница наносила стежки, их описания следуют в произвольном порядке. При этом гарантируется, что узор был получен в результате процесса, описанного в условии задачи.

Следующие q строк описывают фрагменты. Каждое описание содержит четыре целых числа x_1 , y_1 , x_2 и y_2 — координаты левой нижней и правой верхней клетки фрагмента ($1 \leq x_1 \leq x_2 \leq a$; $1 \leq y_1 \leq y_2 \leq b$).

Формат выходных данных

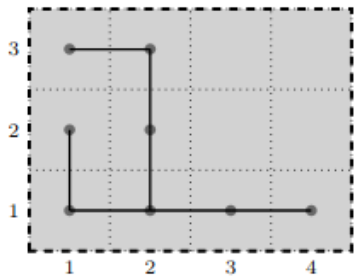
Выходные данные должны содержать q строк, где i -я строка содержит количество связанных частей узора в i -м фрагменте.

Пример

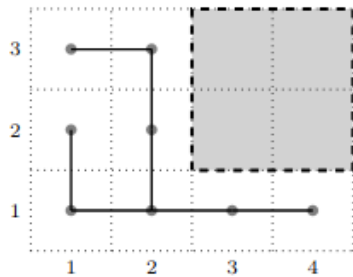
cтандартный ввoд	cтандартный вывoд
4 3	1
8 4	0
v 1 1	1
h 1 1	2
h 2 1	
v 2 1	
v 2 2	
h 1 3	
h 3 1	
1 1 4 3	
3 2 4 3	
3 1 3 1	
1 2 3 3	

Замечание

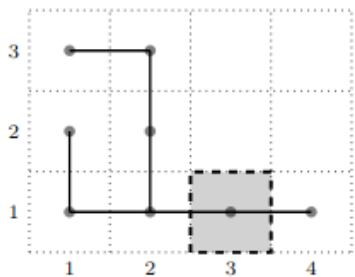
Пояcнение к тecту из условия.



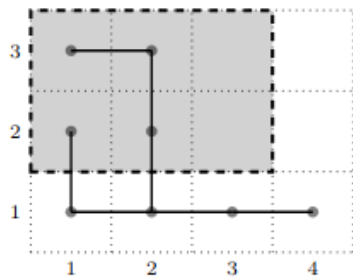
$x_1 = 1, y_1 = 1, x_2 = 4, y_2 = 3$



$x_1 = 3, y_1 = 2, x_2 = 4, y_2 = 3$



$x_1 = 3, y_1 = 1, x_2 = 3, y_2 = 1$



$x_1 = 1, y_1 = 2, x_2 = 3, y_2 = 3$

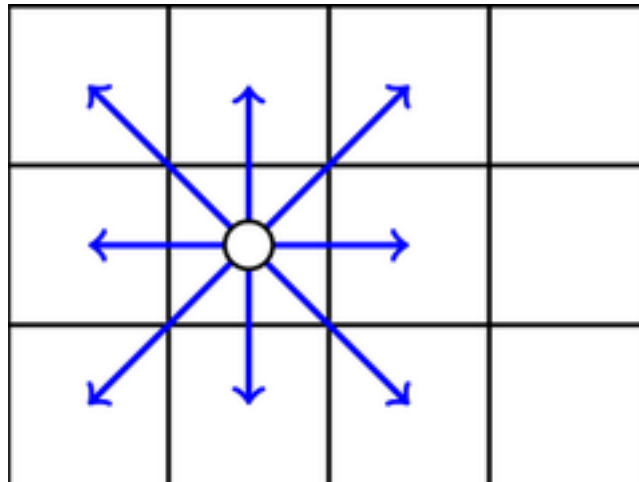
Задача Е. Полезные ископаемые

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

Ведется проект по освоению планеты соседней звездной системы. Для добычи полезных ископаемых планируется направить на планету несколько партий роботов.

Участок поверхности планеты, на котором планируется добывать полезные ископаемые, представляет собой клетчатый прямоугольник размером w на h , клетки участка имеют координаты от $(1, 1)$ до (w, h) . В некоторых клетках участка находятся базы специалистов, в которые могут быть доставлены партии роботов. Всего на участке размещено s баз, и i -я база находится в клетке с координатами (x_i, y_i) .

Каждая партия роботов характеризуется тремя параметрами: j -я партия доставляется на базу b_j , содержит n_j роботов и каждый робот партии обладает мобильностью m_j . Когда партия роботов доставляется на соответствующую базу, каждый робот этой партии перемещается по поверхности планеты от базы до некоторой клетки. Если мобильность робота равна m , он может не более m раз переместиться на одну из восьми соседних клеток, как показано на рис. 1.



После того как роботы из всех доставленных партий размещаются на участке, они активируются и начинают добычу полезных ископаемых. В процессе перемещения в одной клетке может одновременно находиться произвольное количество роботов. Однако после активации в каждой клетке должно находиться не более q роботов.

Руководством проекта получена информация о t партиях роботов, которые могут быть последовательно отправлены на планету. После доставки всех партий роботов, учитывая их ограниченную мобильность, возможна ситуация, что не удастся разместить роботов на участке так, чтобы в каждой клетке оказалось не больше q роботов. Поэтому руководство должно выбрать k первых партий роботов, где $0 \leq k \leq t$, которые будут полностью доставлены на соответствующие базы. После этого, если $k < t$, следует дополнительно принять z из n_{k+1} роботов следующей, $(k + 1)$ -й партии, $0 \leq z < n_{k+1}$.

Все полученные таким образом роботы должны с учетом ограничения на мобильность разместиться на участке таким образом, чтобы в каждой клетке было не более q роботов. После этого они будут активированы и начнут добычу полезных ископаемых. Разумеется, руководство проекта старается максимизировать количество роботов, которые будут доставлены на планету, поэтому, с учетом описанных ограничений, требуется максимизировать k , а затем максимизировать z .

Требуется написать программу, которая по размерам участка, числу q , описанию расположения баз, а также количеству запланированных партий роботов и их описанию определяет максимальное число k — количество партий роботов, и затем — максимальное число z — дополнительное количество роботов из $(k + 1)$ -й партии, чтобы, доставив роботов на планету, их можно было разместить на участке таким образом, чтобы в каждой клетке оказалось не более q роботов.

Формат входных данных

Первая строка входного файла содержит числа w , h , s и q ($1 \leq w, h \leq 10^5$, $1 \leq s \leq 4$, $1 \leq q \leq 100$). Последующие s строк содержат по два целых числа x_i , y_i и описывают базы специалистов ($1 \leq x_i \leq w$, $1 \leq y_i \leq h$).

Следующая строка содержит число t – количество партий роботов ($1 \leq t \leq 100$). Последующие t строк описывают партии роботов и содержат по 3 целых числа: b_j , n_j и m_j ($1 \leq b_j \leq s$, $1 \leq n_j \leq w \cdot h \cdot q$, $0 \leq m_j < \max(w, h)$).

Формат выходных данных

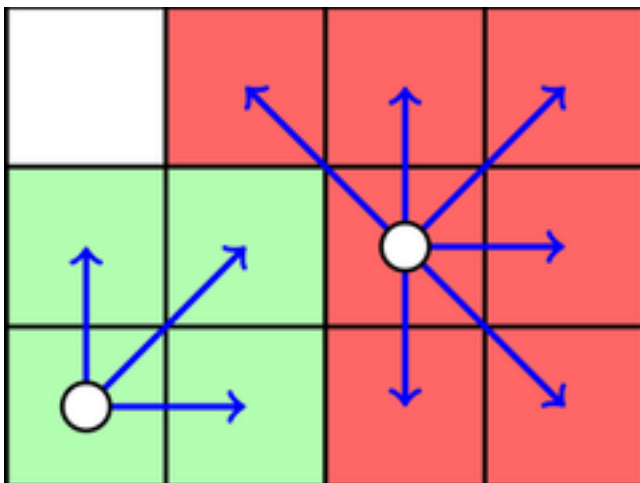
Требуется вывести два числа: k и z , $0 \leq k \leq t$. Если $k = t$, то z должно быть равно 0, иначе должно выполняться условие $0 \leq z < n_{k+1}$.

Пример

стандартный ввод	стандартный вывод
4 3 2 1 1 1 3 2 3 1 4 1 2 9 1 1 12 2	1 7

Замечание

В приведенном примере описания входных данных следует полностью принять первую партию роботов и дополнительно принять 7 роботов из второй партии. На рис. 2 показано, как можно разместить этих роботов на участке, чтобы в каждой клетке было не более одного робота. Базы специалистов показаны кружками. Клетки, в которых окажутся роботы с базы 1, показаны зелёным, а клетки, в которых окажутся роботы с базы 2, показаны красным цветом.



Возможное размещение роботов на участке в данном примере.