

Задача А. Yet Another Tree Problem

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

Дано дерево на n вершинах (дерево — это неориентированный связный граф без циклов). Каждое ребро имеет вес — целое число. Некоторые вершины дерева помечены. Вам нужно реализовать программу, выполняющую следующие операции с этим деревом:

1. Пометить вершину. Гарантируется, что перед этой операцией она была не помечена.
2. Сделать вершину не помеченной. Гарантируется, что до этого она была помечена.
3. Изменить вес ребра.

До выполнения всех операций и после каждой операции ваша программа должна вывести максимальное число x такое, что существует простой путь с **хотя бы двумя** помеченными вершинами на нем и суммарным весом x . Если в дереве нет двух помеченных вершин, выведите строку «BAD».

Формат входных данных

В первой строке заданы два целых числа n и q ($1 \leq n, q \leq 150\,000$), число вершин и число операций соответственно.

В следующей строке находятся n чисел, означающих, помечена ли вершина: 1 означает, что вершина помечена, 0 — не помечена.

В следующих $n - 1$ строках задаются ребра дерева. Для всех i от 2 до n записаны два числа p_i ($1 \leq p_i < i$) и w_i ($-10^9 \leq w_i \leq 10^9$), которые означают, что в дереве есть ребро между вершинами i и p_i веса w_i .

В следующих q строках задаются операции. Каждая операция имеет одну из следующих форм:

- 1 x ($1 \leq x \leq n$): сделать вершину x помеченной,
- 2 x ($1 \leq x \leq n$): сделать вершину x не помеченной,
- 3 $x w$ ($2 \leq x \leq n, -10^9 \leq w \leq 10^9$): сделать вес ребра между вершинами p_x и x равным w .

Формат выходных данных

Выведите $q + 1$ строку. В первой строке выведите ответ до выполнения операций. В каждой следующей строке выведите ответ после выполнения очередной операции.

Система оценки

Обозначим за D максимальное число ребер, выходящих из одной вершины, а за H — максимальное число вершин на простом пути от вершины 1 до какой-то другой вершины.

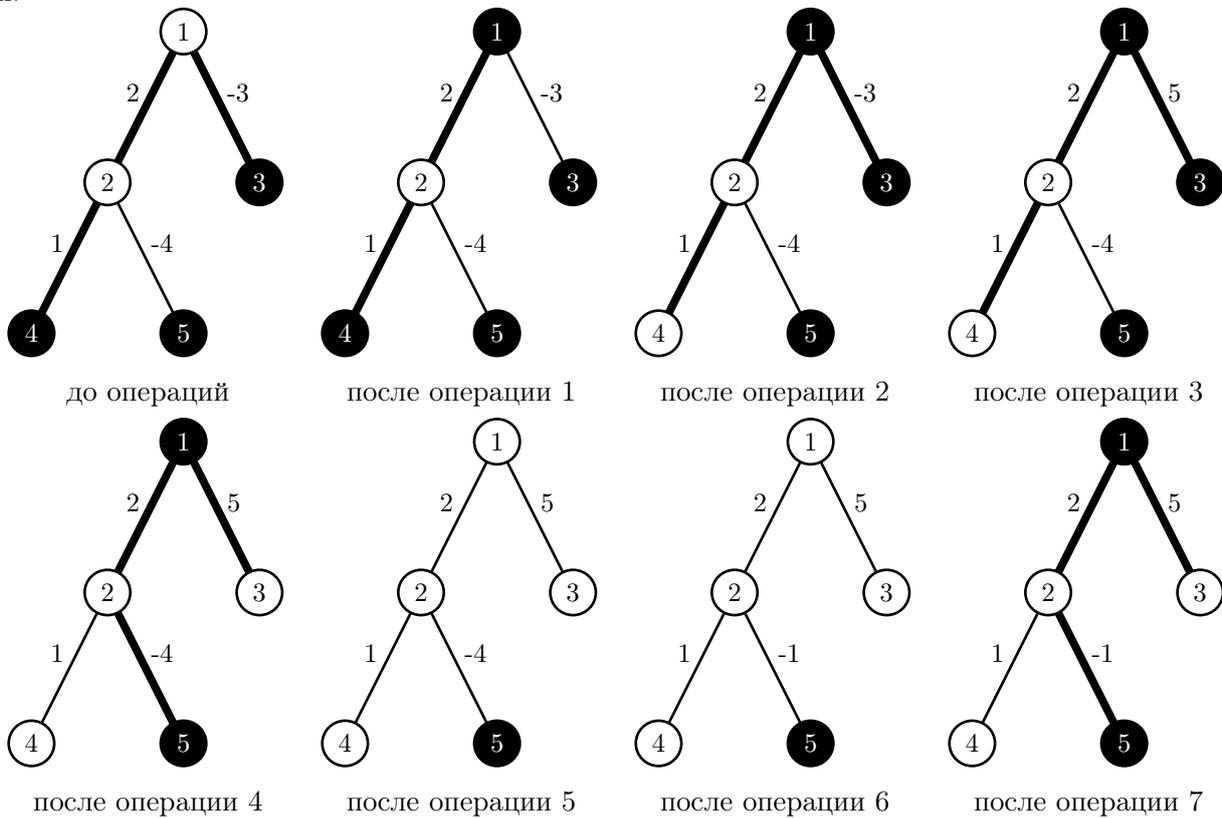
Подзадача	Баллы	Ограничения		
		n	q	Additional
1	3	$n \leq 20$	$q \leq 20$	—
2	3	$n \leq 500$	$q \leq 500$	—
3	3	$n \leq 100$	$q \leq 5000$	—
4	5	$n \leq 5000$	$q \leq 5000$	—
5	7	$n \leq 10^5$	$q \leq 10^5$	$D \leq 30, H \leq 20$
6	20	$n \leq 10^5$	$q \leq 10^5$	$H \leq 20$
7	10	$n \leq 10^5$	$q \leq 10^5$	$p_v = v - 1$ для всех v
8	17	$n \leq 10^5$	$q \leq 10^5$	$w_i \leq 0, w \leq 0$ для всех операций
9	22	$n \leq 50\,000$	$q \leq 50\,000$	—
10	5	$n \leq 10^5$	$q \leq 10^5$	—
11	5	$n \leq 150\,000$	$q \leq 150\,000$	—

Пример

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

Замечание

Иллюстрации к примеру. Жирным выделен путь веса x с хотя бы двумя помеченными вершинами.



Задача В. Два проспекта

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Для того чтобы сделать столицу Берляндии более привлекательным туристическим местом, великий король придумал следующий план: выбрать две улицы города и назвать их проспектами. Естественно, эти проспекты будут объявлены чрезвычайно важными историческими местами, что должно привлечь туристов со всего мира.

Столицу Берляндии можно представить в виде графа, вершинами которого являются перекрестки, а ребрами являются улицы, соединяющие два перекрестка напрямую. Всего в графе n вершин и m ребер, по любой улице можно двигаться в обоих направлениях, от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам, каждая улица соединяет два различных перекрестка, и никакие две улицы не соединяют одинаковую пару перекрестков.

Чтобы снизить поток обычных горожан, перемещающихся по великим проспектам, было решено ввести платный проезд по каждому из них в обе стороны. Теперь за один проезд по проспекту нужно заплатить 1 тугрик. За проезд по остальным улицам платить не нужно.

Аналитики собрали выборку из k горожан, i -му из них надо ездить на работу от перекрестка a_i к перекрестку b_i . После выбора двух проспектов каждый горожанин будет добираться на работу вдоль пути, стоимость которого будет минимальна.

Для того чтобы заработать как можно больше денег, было решено выбрать в качестве двух проспектов две такие улицы, что суммарное количество тугриков, которые заплатят эти k горожан, будет максимально возможным. Помогите королю: по заданной схеме города и выборке горожан найдите, какие две улицы нужно сделать проспектами, и сколько тугриков заплатят горожане при таком выборе.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится два целых числа, первое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Второе число g ($0 \leq g \leq 7$) — номер группы, к которой принадлежит тест. Далее следуют описания наборов входных данных.

В первой строке описания каждого набора входных данных находятся два целых числа n и m ($3 \leq n \leq 500\,000$, $n - 1 \leq m \leq 500\,000$, $m \leq \frac{n(n-1)}{2}$) — количество перекрестков и улиц города.

В следующих m строках содержатся описания улиц, в i -й строке находятся два целых числа s_i и f_i ($1 \leq s_i, f_i \leq n$, $s_i \neq f_i$) — номера перекрестков, которые соединяет i -я улица. Гарантируется, что никакие две улицы не соединяют одну и ту же пару перекрестков, и что от любого перекрестка можно добраться до любого другого, перемещаясь только по улицам.

В следующей строке находится единственное целое число k ($1 \leq k \leq 500\,000$) — количество горожан в выборке.

В следующих k строках содержатся описания горожан, в i -й строке находятся два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — i -й горожанин едет на работу от перекрестка a_i до перекрестка b_i .

Пусть M обозначает сумму значений m по всем наборам входных данных, а K означает сумму значений k по всем наборам входных данных. Гарантируется, что $M, K \leq 500\,000$.

Формат выходных данных

Для каждого набора входных данных в выведите ответ на задачу.

В первой строке ответа данных выведите суммарное количество тугриков, которое заплатят горожане.

Во второй строке ответа выведите два целых числа x_1 и y_1 — номера перекрестков, дорогу между которыми нужно сделать первым проспектом.

В третьей строке ответа выведите два целых числа x_2 и y_2 — номера перекрестков, дорогу между которыми нужно сделать вторым проспектом.

Номера перекрестков, соединенных улицей, можно выводить в произвольном порядке, каждая из двух выведенных улиц должна встречаться среди m улиц города, выбранные улицы должны быть различными.

Пример

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

Система оценки

Тесты к этой задаче состоят из 7 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Школа ЦПМ 2022-2023. 3 группа. Набираем баллы
столичный Центр Помощи Мигрантам, 6 марта 2023

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		n	m	k		
0	0	–	–	–	–	Тесты из условия
1	14	$n \leq 20$	$m \leq 20$	$K \leq 1000$	0	$t \leq 100$
2	11	$n \leq 100$	$M \leq 2000$	$K \leq 2000$	0 – 1	
3	15	$n \leq 2000$	$M \leq 20\,000$	$K \leq 20\,000$	0 – 2	
4	16	–	$M \leq 100\,000$	$K \leq 100\,000$	0 – 3	
5	11	–	–	–	–	$n = m + 1$
6	19	–	–	–	–	Из каждого перекрестка выходит ровно 2 улицы
7	14	–	–	–	0 – 6	Offline-проверка

Задача С. Блогеры-путешественники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Ян и Татьяна решили стать блогерами-путешественниками и публиковать ролики о поездках по городам своей страны.

В стране есть n городов, пронумерованных от 1 до n . Город 1 — столица их страны. Города соединены m двусторонними дорогами, пронумерованными от 1 до m , каждая из которых соединяет два различных города. При этом одну и ту же пару городов могут соединять несколько различных дорог. Из любого города по дорогам можно доехать до любого другого города страны.

Путешественники планируют отправиться из столицы в какой-то другой город, но пока не выбрали в какой. Маршрут путешествия в город k будет состоять из городов s_1, s_2, \dots, s_q и дорог r_1, r_2, \dots, r_{q-1} , таких что:

- $s_1 = 1, s_q = k$;
- дорога r_i соединяет города s_i и s_{i+1} ;
- ребята не проезжают по одной и той же дороге дважды, поэтому все r_i различны. Допускается проезжать несколько раз через один и тот же город, в том числе через город 1, где путешествие начинается, и город k , в котором путешествие заканчивается.

Для каждой дороги Ян и Татьяна посчитали длительность ролика, который получится при съемке путешествия по этой дороге, длительность ролика для дороги с номером i равна t_i .

В процессе путешествия каждый из ребят выберет одну из дорог маршрута и снимет ролик, посвящённый этой дороге. При этом Ян любит снимать короткие ролики, поэтому выберет на маршруте дорогу с наименьшим значением t_i , а Татьяна предпочитает длинные ролики, поэтому выберет дорогу с наибольшим значением t_i .

Суммарная длина двух роликов будет равна $\min_{1 \leq i \leq q-1} t_{r_i} + \max_{1 \leq i \leq q-1} t_{r_i}$.

Ребята планируют выложить ролики на известную платформу, где большей популярностью пользуются короткие ролики, поэтому они хотят минимизировать суммарную длину двух роликов. Чтобы выбрать конечный город и маршрут для путешествия, блогеры хотят для каждого конечного города k подсчитать минимальную по всем возможным маршрутам из города 1 в город k суммарную длину двух роликов.

Формат входных данных

В первой строке даны два целых числа n, m ($2 \leq n \leq 300\,000, 1 \leq m \leq 300\,000$) — количество городов и дорог.

Следующие m строк содержат описания дорог. В i -й из этих строк находятся три целых числа u_i, v_i, t_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 0 \leq t_i \leq 10^9$) — номера городов, соединённых дорогой, и длительность ролика про эту дорогу.

Гарантируется, что по имеющимся дорогам можно проехать из любого города в любой другой, возможно, через другие города.

Формат выходных данных

Для каждого $2 \leq k \leq n$ выведите минимальную суммарную длину роликов Яна и Татьяны для путешествия, заканчивающегося в городе k .

Примеры

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

Замечание

В первом примере возможные оптимальные маршруты:

- $1 \xrightarrow{t=1} 3 \xrightarrow{t=1} 2$. Длина роликов в маршруте $1 + 1 = 2$.
- $1 \xrightarrow{t=1} 3$. Длина роликов в маршруте $1 + 1 = 2$.

Во втором примере возможные оптимальные маршруты:

- $1 \xrightarrow{t=2} 2$. Длина роликов в маршруте $2 + 2 = 4$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=3} 3$. Длина роликов в маршруте $2 + 3 = 5$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=3} 3 \xrightarrow{t=4} 5 \xrightarrow{t=4} 4$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=3} 3 \xrightarrow{t=4} 5$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=3} 3 \xrightarrow{t=4} 5 \xrightarrow{t=4} 4 \xrightarrow{t=4} 6$. Длина роликов в маршруте $2 + 4 = 6$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=8} 1 \xrightarrow{t=6} 7$. Длина роликов в маршруте $2 + 8 = 10$.

В третьем примере возможные оптимальные маршруты:

- $1 \xrightarrow{t=2} 2 \xrightarrow{t=0} 3 \xrightarrow{t=1} 4 \xrightarrow{t=3} 2$. Длина роликов в маршруте $0 + 3 = 3$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=0} 3$. Длина роликов в маршруте $0 + 2 = 2$.
- $1 \xrightarrow{t=2} 2 \xrightarrow{t=0} 3 \xrightarrow{t=1} 4$. Длина роликов в маршруте $0 + 2 = 2$.

Система оценки

Подз.	Баллы	Ограничения			Необх. подзадачи	Информация о проверке
		n	m	дополнительно		
1	9	$n \leq 300\,000$	$m \leq 300\,000$	$m = n - 1$		первая ошибка
2	17	$n \leq 300\,000$	$m \leq 300\,000$	$t_i = 0$ для всех дорог i из города 1		первая ошибка
3	12	$n \leq 300\,000$	$m \leq 300\,000$	$t_i = 10^9$ для всех дорог i из города 1		первая ошибка
4	9	$n \leq 10$	$m \leq 10$	каждая пара городов соединена не более чем одной дорогой		первая ошибка
5	6	$n \leq 20$	$m \leq 20$	каждая пара городов соединена не более чем одной дорогой	4	первая ошибка
6	6	$n \leq 2000$	$m \leq 2000$	$ u_i - v_i = 1$ для всех дорог		первая ошибка
7	9	$n \leq 2000$	$m \leq 2000$		У, 4-6	первая ошибка
8	8	$n \leq 5000$	$m \leq 300\,000$		У, 4-7	только баллы
9	10	$n \leq 300\,000$	$m \leq 300\,000$	для всех a существует дорога между парой городов a и $a + 1$; для любой пары дорог i и j , для которых $ u_i - v_i = 1$ и $ u_j - v_j > 1$ выполнено $t_i \leq t_j$	6	только баллы
10	14	$n \leq 300\,000$	$m \leq 300\,000$		У, 1-9	только баллы

Задача D. Прибавление

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Дан двумерный массив целых чисел a , состоящий из n строк и m столбцов. Изначально все числа в массиве равны нулю. Необходимо обработать q запросов следующего вида:

- 1 r_1 c_1 r_2 c_2 t ($1 \leq r_1 \leq r_2 \leq n$, $1 \leq c_1 \leq c_2 \leq m - t + 1$, $1 \leq t \leq m$). Назовём операцией *прибавления в прямоугольнике*, заданном углами (x_1, y_1) и (x_2, y_2) , где $x_1 \leq x_2$, $y_1 \leq y_2$, процедуру, в результате которой ко всем элементам массива $a_{x,y}$ для которых $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, прибавляется 1.

Необходимо для всех i , таких что $0 \leq i < t$ выполнить операцию прибавления в прямоугольнике, заданном углами $(r_1, c_1 + i)$, $(r_2, c_2 + i)$.

- 2 r_1 c_1 r_2 c_2 ($1 \leq r_1 \leq r_2 \leq n$, $1 \leq c_1 \leq c_2 \leq m$). Необходимо вычислить сумму элементов $a_{x,y}$ в прямоугольнике, заданном углами (r_1, c_1) , (r_2, c_2) . Поскольку данная сумма может быть очень большой, надо вычислить остаток от её деления на 2^{31} .

Формат входных данных

В первой строке заданы три целых числа n , m и q ($1 \leq n, m \leq 10^9$, $1 \leq q \leq 100\,000$) — размеры массива и количество запросов.

В следующих q строках находятся описания запросов в формате, описанном выше.

Формат выходных данных

Для каждого запроса второго типа выведите ответ в отдельной строке. Гарантируется, что хотя бы один запрос имеет тип 2.

Примеры

стандартный ввод	стандартный вывод
1 4 2 1 1 1 1 2 3 2 1 1 1 4	6
4 8 6 1 1 1 2 4 3 1 2 1 3 2 2 1 3 3 4 4 1 2 3 3 4 6 2 1 2 3 5 2 1 5 4 8	5 28 6
6 6 6 1 1 1 3 5 1 1 2 1 4 4 3 2 1 1 6 5 1 2 2 6 5 1 1 2 2 5 6 1 2 2 2 3 4	48 34
13 8 8 2 2 1 8 5 1 4 2 7 8 1 2 1 2 8 4 1 3 2 13 5 2 1 4 2 8 5 3 1 2 2 9 4 1 1 4 2 9 5 2 2 3 2 7 5	0 12 130

Замечание

В первом примере мы имеем дело с одномерным массивом длины 4. Изначально массив равен $(0, 0, 0, 0)$. После первого запроса к элементам на отрезках $[1; 2]$, $[2; 3]$ и $[3; 4]$ прибавляем по 1, и получается массив $(1, 2, 2, 1)$ с суммой элементов 6.

Во втором примере в результате первых трёх запросов происходят следующие преобразования (строки нумеруются снизу вверх):

0	0	0	0	0	0	0	0	0	→	0	0	0	0	0	0	0	0	→	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0		1	2	2	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	→	1	2	3	3	2	1	0	0	→	2	4	4	3	2	1	0	0	0
0	0	0	0	0	0	0	0	0		1	2	3	3	2	1	0	0	→	1	2	3	3	2	1	0	0	0

Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n, m	q		
0	0	–	–	–	Тесты из условия.
1	7	$n, m \leq 100$	$q \leq 100$	0	
2	8	$n, m \leq 700$	$q \leq 700$	0 – 1	
3	10	–	$q \leq 10\,000$	–	$t_i = 1$.
4	15	–	$q \leq 10\,000$	0 – 3	
5	9	–	–	–	В запросах второго типа $(r_{i,1}, c_{i,1}) = (1, 1)$, $(r_{i,2}, c_{i,2}) = (n, m)$.
6	11	$n, m \leq 1000$	–	–	Все запросы типа 2 идут после запросов типа 1, $t_i = 1$.
7	10	$n, m \leq 1000$	–	6	Все запросы типа 2 идут после запросов типа 1.
8	14	–	–	3, 6	$t_i = 1$.
9	16	–	–	0 – 8	

Задача Е. Точки на плоскости

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Даны n точек на плоскости и q запросов следующего вида:

- 1 $x_1 y_1 x_2 y_2$ ($x_1 \neq x_2$ или $y_1 \neq y_2$). Рассмотрим прямую, проходящую через точки (x_1, y_1) и (x_2, y_2) . Существуют две полуплоскости, для которых данная прямая является границей. Рассмотрим ту из них, в которую попадает точка $(x_1 + y_2 - y_1, y_1 + x_1 - x_2)$. Другими словами, если рассмотреть направленный вектор из (x_1, y_1) в (x_2, y_2) , то требуемая полуплоскость будет лежать справа. Вам необходимо проверить, принадлежит ли хотя бы одна из данных точек заданной полуплоскости. Обратите внимание, что прямая принадлежит полуплоскости, поэтому точки, попадающие на прямую, учитываются.
- 2 $x_1 y_1 x_2 y_2$ ($x_1 \neq x_2$ или $y_1 \neq y_2$). Рассмотрим квадрат, противоположными вершинами которого являются точки (x_1, y_1) и (x_2, y_2) . Обратите внимание, что стороны этого квадрата не обязательно параллельны осям координат. Необходимо проверить, есть ли хотя бы одна заданная точка, которая лежит внутри или на границе квадрата.

Формат входных данных

В первой строке заданы два целых числа n и q ($1 \leq n, q \leq 100\,000$) — количество точек и количество запросов.

В следующих n строках содержатся пары целых чисел x_i, y_i ($1 \leq x_i, y_i \leq 10^8$), описывающие координаты точек на плоскости.

В следующих q строках заданы запросы в формате, описанном выше. ($1 \leq x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2} \leq 10^8$)

Формат выходных данных

Для каждого запроса выведите «Yes», если в заданном объекте лежит хотя бы одна точка и «No» — иначе.

Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

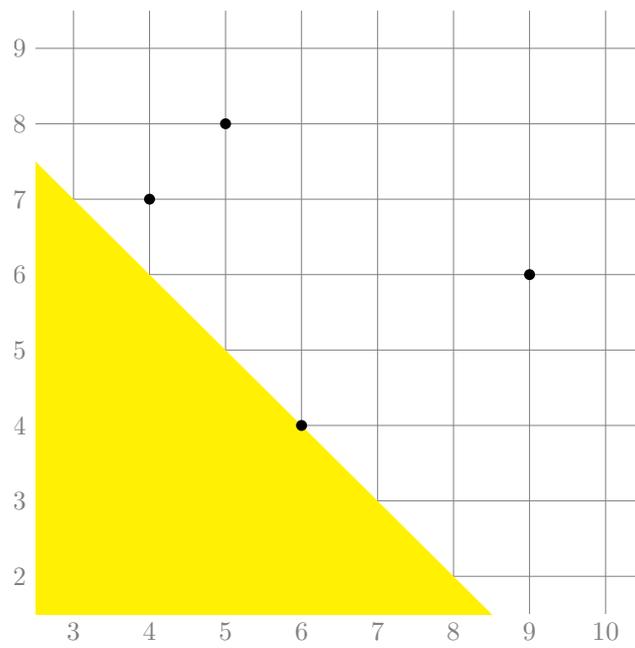
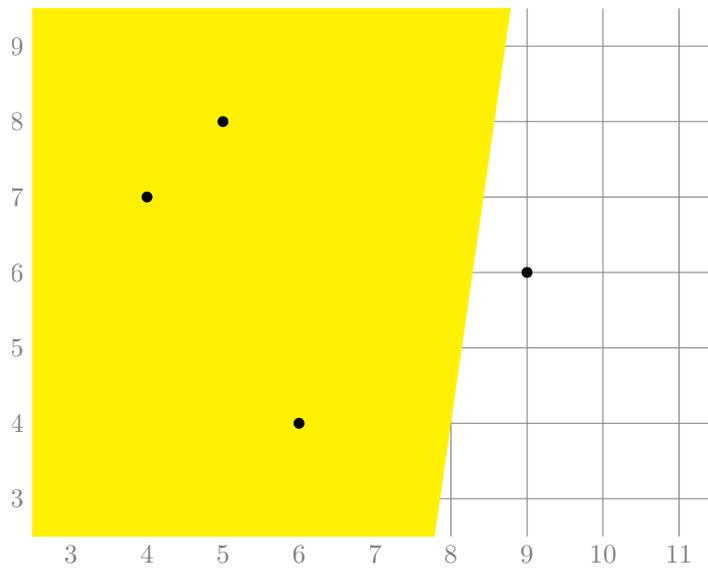
Группа	Баллы	Дополнительные ограничения		Необх. группы	Комментарий
		n	q		
0	0	–	–	–	Тесты из условия.
1	11	$n \leq 1000$	$q \leq 1000$	–	Для всех i верно, что $t_i = 1$.
2	12	–	–	1	Для всех i верно, что $t_i = 1$.
3	5	$n \leq 1000$	$q \leq 1000$	–	Для всех i верно, что $t_i = 2$ и $x_{i,1} - x_{i,2} = y_{i,1} - y_{i,2}$.
4	16	–	–	3	Для всех i верно, что $t_i = 2$ и $x_{i,1} - x_{i,2} = y_{i,1} - y_{i,2}$.
5	12	$n \leq 1000$	$q \leq 1000$	0, 1, 3	
6	17	$n \leq 30\,000$	$q \leq 30\,000$	0, 1, 3, 5	
7	9	$n \leq 60\,000$	$q \leq 60\,000$	0, 1, 3, 5, 6	
8	18	–	–	0 – 7	

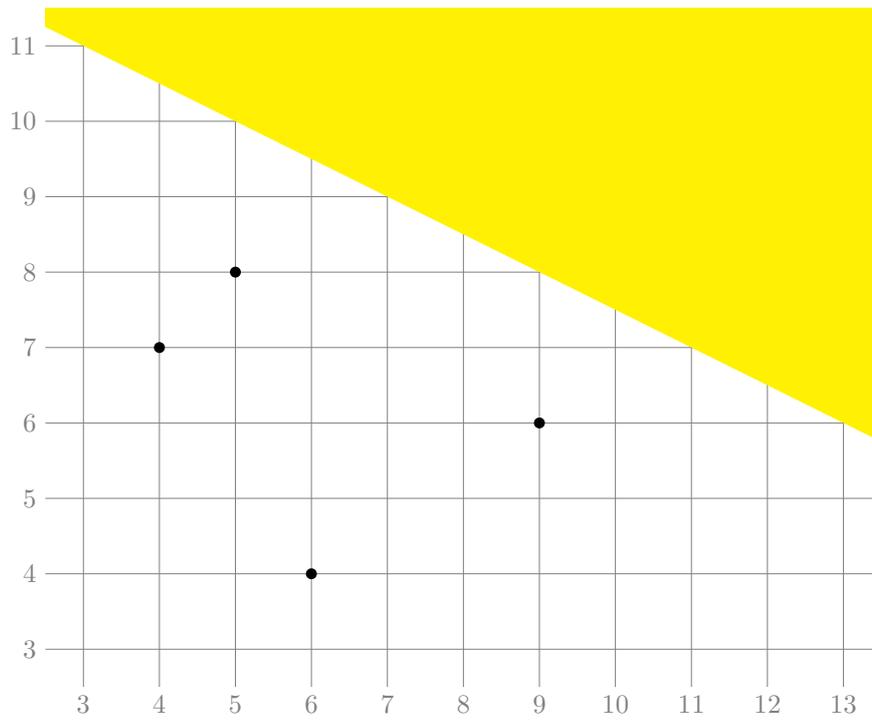
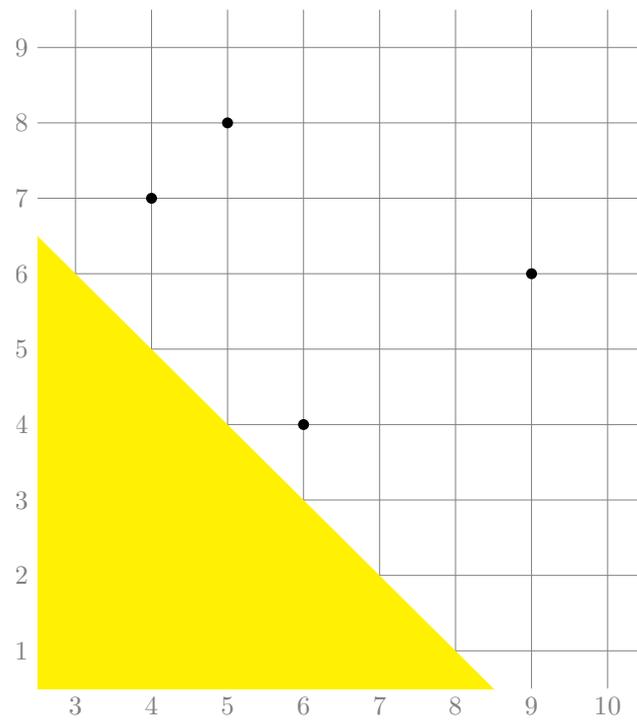
Примеры

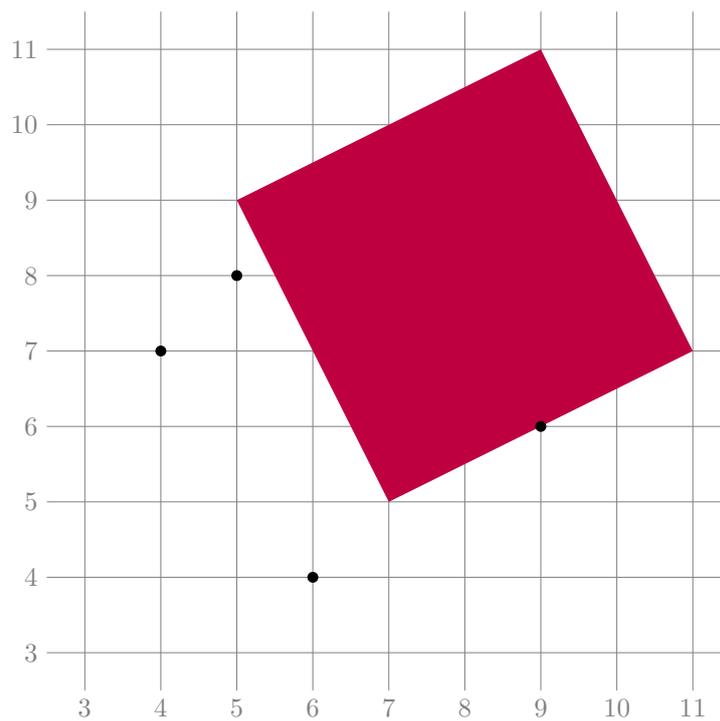
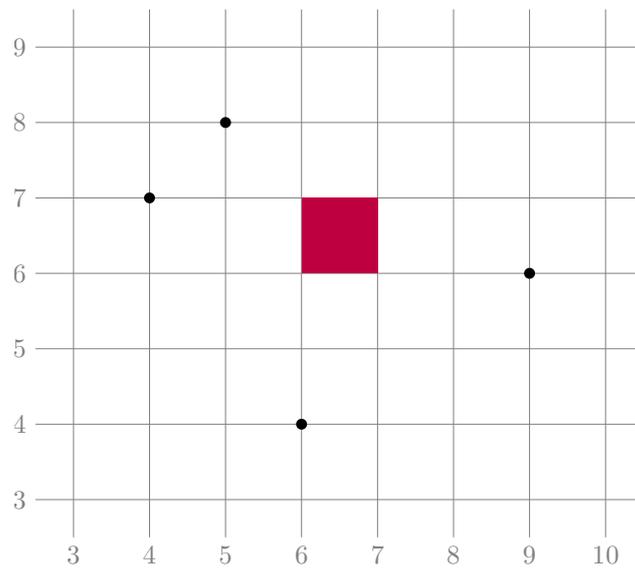
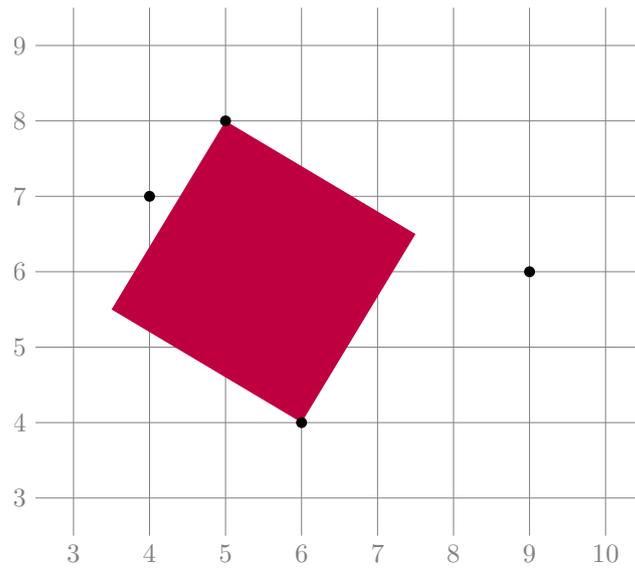
стандартный ввод	стандартный вывод
4 4	Yes
4 7	Yes
5 8	No
6 4	No
9 6	
1 9 11 8 4	
1 3 7 8 2	
1 3 6 8 1	
1 13 6 3 11	
4 8	Yes
4 7	Yes
5 8	No
6 4	No
9 6	Yes
1 9 11 8 4	No
1 3 7 8 2	Yes
1 3 6 8 1	Yes
1 13 6 3 11	
2 6 4 5 8	
2 6 6 7 7	
2 7 5 9 11	
2 5 3 6 6	

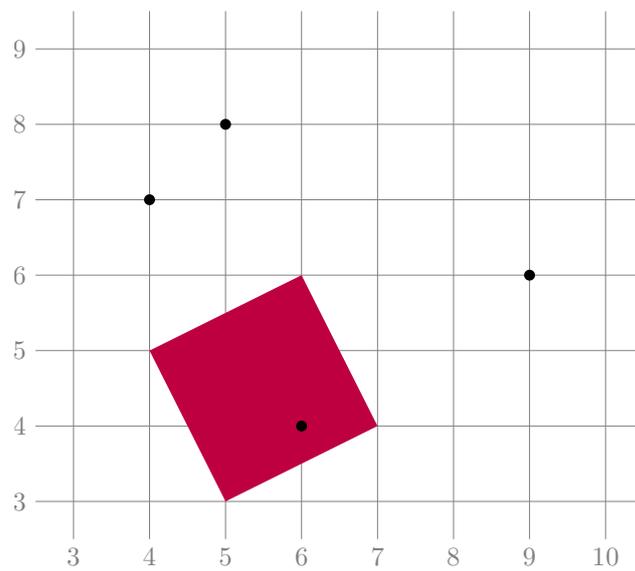
Замечание

Ниже приведены точки и области для каждого из случаев второго примера









Задача F. Вставить текст

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Алиса и Аня работают копирайтерами. Недавно им пришёл заказ: нужно написать k текстов (строк) на схожую тематику. Девочки сразу приступили к работе и быстро получили k строк S_1, \dots, S_k , каждая из которых имеет длину не более m и состоит из строчных латинских букв, при чем длина S_1 оказался в точности равна m .

Девочки уже давно работают вместе, и у них есть простой способ проверить оригинальность своей работы. Оригинальность оценивается с помощью подстрок полученных текстов. Алиса и Аня считают некоторую строку неоригинальной, если и она, и строка, получающаяся из данной разворотом, встречаются в некоторых из их текстов в качестве подстрок на одних и тех же позициях.

Чтобы ускорить проверку, они разбивают строки на блоки. Разбиение на t блоков задается последовательностью a_0, a_1, \dots, a_t , где $a_0 = 0$, $a_t = m$ и $a_{i-1} < a_i$ для любого $1 \leq i \leq t$. Тогда i -м блоком называется отрезок целых чисел $[a_{i-1} + 1; a_i]$. Блок $[a_{i-1} + 1; a_i]$ называется интересным, если существует неоригинальная строка, встречающаяся в текстах ровно на позициях, задаваемых данным блоком. Иными словами, блок — интересный, если для каких-то строк S_l и S_r (возможно, $l = r$) верно, что $|S_l|, |S_r| \geq a_i$, и строка $S_{l, a_{i-1}+1} S_{l, a_{i-1}+2} \dots S_{l, a_i}$ совпадает со строкой $S_{r, a_i} S_{r, a_i-1} \dots S_{r, a_{i-1}+1}$, где $S_{t,j}$ — j -й слева символ строки S_t .

Например, для текстов [abba, ba] последовательности $(0, 1, 4)$ и $(0, 1, 2, 3, 4)$ задают корректные разбиения, а последовательности $(1, 2, 3)$ и $(0, 1, 1, 4)$ — нет. При этом для разбиения $(0, 2, 4)$ первый блок $[1; 2]$ является интересным, поскольку $S_{1,1} S_{1,2} = S_{2,2} S_{2,1} = ab$, а второй блок $[3; 4]$ — нет, поскольку для единственной возможной пары номеров $l = r = 1$ строки $S_{1,3} S_{1,4} = ba$ и $S_{1,4} S_{1,3} = ab$ не совпадают.

Разбиение называется интересным, если каждый блок в этом разбиении интересный. Алиса и Аня хотят найти интересное разбиение текстов на минимальное число блоков, чтобы измерить оригинальность работы. Девочки постарались, чтобы условие этой задачи было современным и прошло тест Бекдел, поэтому теперь помогите им и напишите для них программу, измеряющую оригинальность!

Заметим, что искомая величина корректно определена, так как, разбив строки на m блоков длины 1, мы получим интересное разбиение (в каждом блоке будет достаточно взять $l = r = 1$).

Формат входных данных

Во первой строке вводятся три целых числа t , k и m ($1 \leq k \leq 200\,000$, $1 \leq m \leq 500\,000$) — номер группы, к которой относится данный тест, общее число текстов и длина первого текста.

В i -й из следующих k строк вводится S_i — i -й текст, состоящий из строчных букв латинского алфавита.

Гарантируется, что $|S_1| = m$, $|S_i| \leq m$ для любого $i > 1$, и суммарная длина всех строк не превосходит 500 000.

Формат выходных данных

Во первой строке выведите единственное целое число t — минимальное число блоков в интересном разбиении.

Во второй строке через пробел выведите возрастающую последовательность из $t - 1$ целого числа a_1, \dots, a_{t-1} — номеров правых границ всех блоков, кроме последнего (само разбиение имеет вид $[1; a_1], [a_1 + 1; a_2], \dots, [a_{t-1} + 1; m]$).

Примеры

стандартный ввод	стандартный вывод
0 2 6 abcded cba	2 3
0 6 7 poggers sus amogus tokyo ghoul sodluv	4 3 5 6

Замечание

В первом примере вторая строка `cba` при перевороте совпадает с первыми тремя символами первой строки `abcded`. Оставшиеся же символы `ded` образуют палиндром, т.е. эта строка совпадает с собой же перевёрнутой. Поэтому мы можем разбить строки на два блока $[1; 3]$ и $[4; 6]$. Легко видеть, что на меньшее число блоков разбить нельзя, ведь `abcded` — не палиндром.

Во втором примере в первом блоке $[1; 3]$ можно выбрать строку-палиндром `sus`, во втором блоке кусочки пятой (`ghoul`) и шестой (`sodluv`) строк, совпадающих друг с другом при перевороте. В третьем и четвертом блоках выбираем по одной букве из любой строки. Можно показать, что на меньшее число блоков разбить строки нельзя.

Система оценки

Тесты к этой задаче состоят из десяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов всех необходимых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Обозначим суммарную длину всех строк как L .

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		k	m	L		
0	0	–	–	–	–	Тесты из условия.
1	7	$k \leq 100$	$m \leq 10$	$L \leq 1000$	0	
2	8	$k \leq 20$	$m \leq 100$	$L \leq 2000$	0	
3	10	$k \leq 200$	$m \leq 200$	$L \leq 40\,000$	0 – 2	
4	12	$k \leq 200$	$m \leq 2000$	$L \leq 400\,000$	0 – 3	
5	9	–	–	–	–	Гарантируется, что в ответе не больше двух блоков.
6	11	$k = 1$	$m \leq 50\,000$	$L \leq 50\,000$	–	
7	11	$k = 1$	–	–	6	
8	10	$k = 2$	–	–	–	
9	9	–	$m \leq 200\,000$	$L \leq 200\,000$	0 – 3, 6	
10	13	–	–	–	0 – 9	Offline-проверка.