Задача Wool. Сильная связность наносит ответный удар

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Вершины u и v ориентированного графа называются сильно связными, если в графе существует путь из u в v и путь из v в u. Заметим, что если u и v, а также v и w сильно связны, то u и w сильно связны. Поэтому вершины графа разбиваются на множества — компоненты сильной связности. Вершина, принадлежащая компоненте сильной связности, сильно связна со всеми вершинами компоненты (в том числе с собой) и не сильно связна со всеми остальными вершинами.

На занятии по графам Алиса нарисовала на доске ориентированный граф на n вершинах, а также выделила его компоненты сильной связности. Во время перерыва Боб решил разыграть Алису и стереть направления на некоторых рёбрах графа. При этом он хочет, чтобы после перерыва Алиса смогла однозначно восстановить стёртые направления по остальным рёбрам и разбиению на компоненты сильной связности.

Помогите Бобу, сообщив, на каком максимальном количестве рёбер графа он может стереть направления, а также скольким количеством способов он может это сделать.

Более формально, найдите максимальный размер подмножества рёбер графа A, обладающего следующим свойством: если стереть направления рёбер в множестве A, то на основе информации о старых компонентах сильной связности и направлениях рёбер не из множества A можно единственным образом восстановить направления рёбер из множества A так, чтобы компоненты сильной связности остались прежними.

Поскольку количество таких максимальных подмножеств может быть очень большим, выведите его по модулю $10^9 + 7$.

Решения, корректно определяющие максимальный размер множества A, но неправильно предъявляющие количество таких множеств, будут получать частичные баллы.

Формат входных данных

Первая строка содержит три целых числа n, m и g ($2 \leqslant n \leqslant 2000$, $1 \leqslant m \leqslant 2000$, $0 \leqslant g \leqslant 7$) — количество вершин и рёбер графа соответственно, а также номер группы тестов.

Следующие m строк содержат по два целых числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$) — номера вершин, являющихся началом и концом i-го ребра.

Гарантируется, что для любых $1 \leqslant i, j \leqslant n$ $(u_i, v_i) \neq (u_j, v_j)$ и $(u_i, v_i) \neq (v_j, u_j)$, то есть любые две вершины соединены не более чем одним ребром, вне зависимости от направления.

Формат выходных данных

В первой строке выведите единственное число — размер максимального подмножества рёбер, на котором можно стереть направления.

Во второй строке выведите единственное число — количество таких подмножеств по модулю $10^9 + 7$. Если вы не хотите предъявлять количество таких подмножеств, то во второй строке, вместо этого, выведите любое число от -1 до $10^9 + 6$. В этом случае ваше решение получит часть баллов за тест.

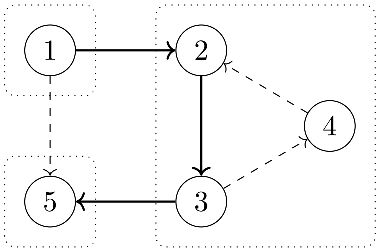
Обратите внимание, что отсутствие какого либо числа во второй строке приводит к вердикту «Неправильный ответ» и нулю баллов за этот тест, несмотря на правильность размера подмножества.

Пример

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

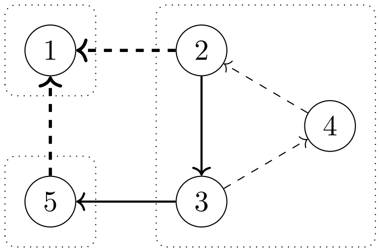
Пояснение

Граф из примера с выделенными компонентами сильной связности:



На рёбрах, выделенных пунктиром, можно стереть направления. Действительно, ребро $(1, 5)$ не может быть ориентировано в другую сторону, потому что иначе вершины 1, 2, 3 и 5 будут лежать в одной компоненте сильной связности. Рёбра $(3, 4)$ и $(4, 2)$ не могут быть ориентированы иначе, потому что тогда вершины 2, 3 и 4 не будут лежать в одной компоненте сильной связности.

Рассмотрим некорректный способ выбрать подмножество рёбер:



Здесь нельзя стереть направления на рёбрах, выделенных жирным пунктиром. Например, если ориентировать рёбра $(1, 2)$ и $(1, 5)$ в другую сторону, получится граф с таким же разбиением на компоненты сильной связности.

Можно показать, что нельзя стереть направления на 4 рёбрах, поэтому ответ — 3.

Система оценки

Тесты к этой задаче состоят из семи групп. Правила, по которым выставляются баллы за подгруппы описаны ниже. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Если во всех тестах группы правильно посчитан размер максимального подмножества рёбер, на котором можно стереть направления и количество таких подмножеств, за группу выставляется полный балл.

В противном случае, если во всех тестах группы правильно посчитан размер максимального подмножества рёбер, на котором можно стереть направления, а количество оказалось неверным хотя-бы в одном тесте группы, за группу выставляется частичный балл. Обратите внимание, что группы, зависящие от неё, в таком случае будут запущены для тестирования и даже могут быть оценены в полный балл.

Группа	Баллы		Доп. ограничения		Необх. группы	Комментарий
	Частичные	Полные	n	m		
0	0	0	–	–	–	Тесты из условия.
1	7	11	$n \leqslant 14$	$m \leqslant 14$	0	
2	5	9	$n \leqslant 20$	$m \leqslant 20$	0, 1	
3	8	12	–	–	–	$u_i < v_i$, для всех $1 \leqslant i \leqslant n-1$ есть ребро $(i, i+1)$
4	8	13	–	–	3	$u_i < v_i$
5	12	20	–	–	–	для всех $1 \leqslant i \leqslant n-1$ есть ребро $(i, i+1)$, есть ребро $(n, 1)$
6	13	21	–	–	5	граф состоит из одной компоненты сильной связности
7	8	14	–	–	0 – 6	

Задача Sand. Лучший бегун

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

На стадионе есть n беговых дорожек с длинами a_1, a_2, $\ldots$, a_n. Также есть m бегунов, i-й бегун находится в начале дорожки b_i.

Все бегуны будут тренироваться в течение T секунд. Тренировка бегуна выглядит так:

Пусть в текущий момент бегун находится в начале дорожки i. Он пробегает до конца текущей дорожки за a_i секунд. Далее он может либо моментально переместиться в начало текущей дорожки, либо в начало $(i-1)$-й дорожки (если $i > 1$), либо в начало $(i+1)$-й дорожки (если $i < n$). После этого он продолжает бег с дорожки, на которую переместился. Как только длительность тренировки достигает T секунд, он завершает тренировку.

Назовём лучшим того бегуна, который пробежит наибольшее количество **полных** дорожек за время тренировки (таких бегунов может быть несколько). Определите, сколько дорожек пробежит лучший бегун.

Формат входных данных

Первая строка содержит три целых числа n, m и T ($1 \leqslant m \leqslant n \leqslant 300\,000$, $1 \leqslant T \leqslant 10^9$) — количество дорожек, количество бегунов и длительность тренировки.

Вторая строка содержит n целых чисел a_1, a_2, $\ldots$, a_n ($1 \leqslant a_i \leqslant 10^9$) — длины дорожек.

Третья строка содержит m целых чисел b_1, b_2, $\ldots$, b_m ($1 \leqslant b_1 < b_2 < \ldots < b_m \leqslant n$) — номера дорожек, с которых начинают бегуны.

Формат выходных данных

Выведите единственное число — максимальное количество полных дорожек, которое может пробежать за время тренировки один из бегунов.

Примеры

ввод	вывод

`5 3 10`
`4 5 2 7 1`
`1 2 4`
	`4`

`4 2 11`
`4 5 7 10`
`2 3`
	`2`

Пояснение

В первом примере больше всех дорожек может пробежать бегун, начинающий на дорожке 4: он должен пробежать дорожку 4, после переместиться на дорожку 5 и пробежать её 3 раза.

Во втором примере бегун, начинающий на дорожке 2, может пробежать вторую дорожку 2 раза.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, что прохождение тестов из условия не требуется для некоторых групп. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		n	T	a_i		
0	0	–	–	–	–	Тесты из условия.
1	23	$n \leqslant 1000$	–	–	0	
2	10	–	–	–	–	$a_i \leqslant a_{i+1}$ для всех $1 \leqslant i < n$
3	16	–	$T \leqslant 20$	–	0	
4	19	–	–	$a_i \leqslant 20$	0	
5	11	–	–	–	–	$m = n$
6	21	–	–	–	0 – 5	

Задача Glass. Переворот карт

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Петя и Вася купили новую карточную игру «Переворот». В наборе есть n двусторонних и m односторонних карт:

1. **Двусторонняя карта.** На лицевой стороне карты записано число a_i, а на обратной — b_i.
2. **Односторонняя карта.** На лицевой стороне карты записано одно число c_i.

Все числа, записанные на картах со всех сторон, различны. Изначально все карты выкладываются на стол лицевой стороной вверх. В свой ход можно сделать ровно одно из двух действий:

1. Убрать со стола карту, на которой записано наименьшее число из оставшихся.
2. Если карта с наименьшим числом двусторонняя и лежит лицевой стороной вверх, то её можно перевернуть.

Выигрывает тот, кто убрал со стола последнюю карту. Определите победителя в данной игре, если первым ходит Петя.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leqslant n, m \leqslant 500\,000$) — количество двусторонних и односторонних карт соответственно.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 2 \cdot n + m$) — числа, записанные на лицевой стороне двусторонних карт.

Третья строка содержит n целых чисел $b_1, b_2, \ldots, b_n$ ($1 \leqslant b_i \leqslant 2 \cdot n + m$) — числа, записанные на обратной стороне двусторонних карт.

Четвёртая строка содержит m целых чисел $c_1, c_2, \ldots, c_m$ ($1 \leqslant c_i \leqslant 2 \cdot n + m$) — числа, записанные на односторонних картах.

Гарантируется, что каждое число от 1 до $2 \cdot n + m$ встречается ровно в одном из массивов a, b или c.

Формат выходных данных

Выведите «First» (без кавычек), если в данной игре побеждает Петя, и «Second» (без кавычек), если побеждает Вася.

Примеры

ввод	вывод
2 1 5 3 1 2 4	First
1 2 2 3 4 1	Second

Пояснение

В первом примере изначально на столе лежат карты 3, 4 и 5. Чтобы победить, Петя своим ходом выкидывает карту 3, после чего Вася обязан выкинуть карту 4, так как она односторонняя. Наконец, Петя сбрасывает карту 5, а значит для хода Васи карт не останется, и Петя выиграет.

Во втором примере изначально на столе лежат карты 1, 2 и 4. Петя обязан сбросить первую карту, так как она односторонняя. Затем, чтобы победить, Вася переворачивает карту 2, а значит на столе будут лежать две карты 3 и 4. Петя обязан сбросить перевёрнутую карту под номером 3, после чего Вася сбросит карту 4. Так как карты закончатся, то Вася одержит победу.

Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих 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. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	m		
0	0	–	–	–	Тесты из условия.
1	12	$n \leqslant 20$	$m \leqslant 10$	0	
2	13	$n \leqslant 20$	–	0, 1	
3	9	–	–	–	$a_i > b_i$
4	10	–	–	–	$max_{i=1}^{n}(a_i) < min_{i=1}^{n}(b_i)$
5	6	–	–	–	Отрезки $[min(a_i, b_i);\ max(a_i, b_i)]$ не пересекаются.
6	11	$n \leqslant 200$	$m \leqslant 200$	–	Отрезки $[min(a_i, b_i);\ max(a_i, b_i)]$ вложены или не пересекаются.
7	14	–	–	5, 6	Отрезки $[min(a_i, b_i);\ max(a_i, b_i)]$ вложены или не пересекаются.
8	13	$n \leqslant 5000$	$m \leqslant 5000$	0, 1, 6	
9	12	–	–	0 – 8	**Offline-проверка.**

Задача Clay. Порядковая статистика

Имя входного файла:	`input.txt` или стандартный поток ввода
Имя выходного файла:	`output.txt` или стандартный поток вывода
Ограничение по времени:	4 секунды
Ограничение по памяти:	1024 мегабайта

Вам дан массив $a_1, a_2, \ldots, a_n$, состоящий из целых чисел, а также целые числа k и m. С массивом m раз производится следующая операция:

- Выбрать i_1, i_2, …, i_k — позиции k наибольших элементов в массиве a. При равенстве двух элементов большим считается элемент, стоящий в массиве на меньшей позиции.
- Уменьшить $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ на 1.

Для x от 1 до n, обозначим через $F_{m,k}(x)$ значение x-й порядковой статистики в массиве, получаемом из a применением m раз операции с заданным параметром k. Для x от 1 до n, x-й порядковой статистикой массива $a_1, a_2, \ldots, a_n$ называется элемент, который будет стоять на позиции x, если массив a отсортировать в порядке неубывания.

Для всех l, r, таких, что $1 \leqslant l \leqslant r \leqslant n$, обозначим через $S_{m,k}(l, r)$ сумму $F_{m,k}(x)$ по всем целым x от l до r. Более формально:

$$S_{m,k}(l, r) = \sum_{x=l}^{r} F_{m,k}(x)$$

Вам даны целые числа m_0 и k_0. Вы должны вычислить для всех x от 1 до n значения $F_{m_0,k_0}(x)$.

После этого вы должны обработать q запросов. j-й запрос ($1 \leqslant j \leqslant q$) может быть одного из трех типов:

1. Вычислить значение $F_{m_j,k_j}(x_j)$.
2. Изменить значение a_{p_j} на v_j.
3. Вычислить значение $S_{m_j,k_j}(l_j, r_j)$.

Все вычисления функций F и S производятся каждый раз независимо и не меняют массив. Все изменения массива в запросах второго типа сохраняются для следующих запросов.

Формат входных данных

Первая строка содержит четыре целых числа n, m_0, k_0 и q ($1 \leqslant n \leqslant 200\,000$, $0 \leqslant m_0 \leqslant 10^9$, $1 \leqslant k_0 \leqslant n$, $0 \leqslant q \leqslant 200\,000$) — длина массива a; количество операций; количество наибольших элементов, уменьшаемых на каждой операции, и количество запросов.

Вторая строка содержит n целых чисел $a_1, a_2, \ldots, a_n$ ($-10^9 \leqslant a_i \leqslant 10^9$, $1 \leqslant i \leqslant n$) — элементы массива a.

В следующих q строках заданы запросы. В j-й из них в начале содержится число t_j ($1 \leqslant t_j \leqslant 3$) — тип j-го запроса.

- Если $t_j = 1$, то далее строка содержит три целых числа m_j, k_j и x_j ($0 \leqslant m_j \leqslant 10^9$, $1 \leqslant k_j, x_j \leqslant n$) — параметры запроса первого типа.
- Если $t_j = 2$, то далее строка содержит два целых числа p_j и v_j ($1 \leqslant p_j \leqslant n$, $-10^9 \leqslant v_j \leqslant 10^9$) — параметры запроса второго типа.
- Если $t_j = 3$, то далее строка содержит четыре целых числа m_j, k_j, l_j и r_j ($0 \leqslant m_j \leqslant 10^9$, $1 \leqslant k_j, l_j, r_j \leqslant n$, $l_j \leqslant r_j$) — параметры запроса третьего типа.

Формат выходных данных

В первой строке выведите n целых чисел $F_{m_0,k_0}(1), F_{m_0,k_0}(2), \ldots, F_{m_0,k_0}(n)$.

Далее, для каждого запроса первого типа выведите в отдельной строке значение $F_{m_j,k_j}(x_j)$, а для каждого запроса третьего типа значение $S_{m_j,k_j}(l_j, r_j)$ — ответ на j-й запрос.

Пример

ввод	вывод
`8 3 2 16`	`-1 -1 0 1 1 1 1 2`
`3 1 2 -1 0 2 -1 4`	`2`
`3 3 2 2 6`	`1`
`1 3 2 4`	`-4`
`3 4 5 3 5`	`-1`
`1 4 5 6`	`-1`
`2 5 -1`	`-1`
`2 6 3`	`1`
`1 3 2 1`	`2`
`1 3 2 3`	`3`
`1 3 2 4`	`7`
`1 3 2 8`	`8`
`1 0 5 6`	`4`
`2 1 5`	`2`
`3 1 3 7 8`	
`3 2 3 5 8`	
`3 3 3 4 7`	
`3 4 3 4 7`	

Пояснение

В примере $n = 8$, $m_0 = 3$, $k_0 = 2$, $q = 16$. Изначально массив a равен $[3, 1, 2, -1, 0, 2, -1, 4]$. Посмотрим, как массив будет меняться, если применить к нему m_0 раз операцию с параметром k_0:

1. Массив равен $[3, 1, 2, -1, 0, 2, -1, 4]$. Два наибольших элемента стоят на позициях 1 и 8. Они уменьшаются на 1, после чего массив становится равен $[2, 1, 2, -1, 0, 2, -1, 3]$.
2. Массив равен $[2, 1, 2, -1, 0, 2, -1, 3]$. Два наибольших элемента стоят на позициях 1 и 8. Они уменьшаются на 1, после чего массив становится равен $[1, 1, 2, -1, 0, 2, -1, 2]$.
3. Массив равен $[1, 1, 2, -1, 0, 2, -1, 2]$. Два наибольших элемента стоят на позициях 3 и 6. Они уменьшаются на 1, после чего массив становится равен $[1, 1, 1, -1, 0, 1, -1, 2]$.

Получаем, что после применения 3 раза операции с параметром 2 к массиву a он становится равен $[1, 1, 1, -1, 0, 1, -1, 2]$. Если этот массив отсортировать, то получится массив $[-1, -1, 0, 1, 1, 1, 1, 2]$. Получаем, что порядковые статистики равны $F_{3,2}(1) = -1$, $F_{3,2}(2) = -1$, $F_{3,2}(3) = 0$, $F_{3,2}(4) = 1$, $F_{3,2}(5) = 1$, $F_{3,2}(6) = 1$, $F_{3,2}(7) = 1$, $F_{3,2}(8) = 2$.

В примере требуется обработать 16 запросов, разберем подробно первые 10 из них:

1. Первый запрос имеет тип $t_1 = 3$, параметры $m_1 = 3$, $k_1 = 2$, $l_1 = 2$, $r_1 = 6$ и требует вычислить значение $S_{3,2}(2, 6)$. Ранее мы уже вычислили значения $F_{3,2}(x)$ для x от 1 до 8, откуда находим ответ на запрос:
$$S_{3,2}(2, 6) = F_{3,2}(2) + F_{3,2}(3) + F_{3,2}(4) + F_{3,2}(5) + F_{3,2}(6) = (-1) + 0 + 1 + 1 + 1 = 2$$
2. Второй запрос имеет тип $t_2 = 1$, параметры $m_2 = 3$, $k_2 = 2$, $x_2 = 4$ и требует вычислить значение $F_{3,2}(4)$. Его мы уже вычисляли, оно равно 1.
3. Третий запрос имеет тип $t_3 = 3$, параметры $m_3 = 4$, $k_3 = 5$, $l_3 = 3$, $r_3 = 5$ и требует вычислить значение $S_{4,5}(3, 5)$, то есть посчитать сумму порядковых статистик с третьей по пятую в массиве, полученном из a применением $m_3 = 4$ раза операции с параметром $k_3 = 5$. На момент третьего запроса массив a равен $[3, 1, 2, -1, 0, 2, -1, 4]$. Пять наибольших элементов стоят на позициях $1, 2, 3, 6, 8$. Уменьшим их на 1, получим массив $[2, 0, 1, -1, 0, 1, -1, 3]$. Применим операцию еще три раза, получим массив $[-1, -2, -2, -2, -1, -1, -1, 0]$. После сортировки он равен $[-2, -2, -2, -1, -1, -1, -1, 0]$. Получаем, что ответ на запрос равен
$$S_{4,5}(3, 5) = F_{4,5}(3) + F_{4,5}(4) + F_{4,5}(5) = (-2) + (-1) + (-1) = -4$$

4. Четвертый запрос имеет тип $t_4 = 1$, параметры $m_4 = 4$, $k_4 = 5$, $x_4 = 6$. После применения четырех операций с параметром 5 и сортировки массив a будет равен $[-2, -2, -2, -1, -1, -1, -1, 0]$, поэтому шестая порядковая статистика равна -1.

5. Пятый запрос имеет тип $t_5 = 2$, параметры $p_5 = 5$ и $v_5 = -1$. Он меняет значение a_5 на -1, после чего массив a становится равен $[3, 1, 2, -1, -1, 2, -1, 4]$.

6. Шестой запрос имеет тип $t_6 = 2$, параметры $p_6 = 6$ и $v_6 = 3$. Он меняет значение a_6 на 3, после чего массив a становится равен $[3, 1, 2, -1, -1, 3, -1, 4]$.

7. Седьмой запрос требует найти значение $F_{3,2}(1)$. Массив a на момент седьмого запроса равен $[3, 1, 2, -1, -1, 3, -1, 4]$. После применения 3 раза операции с параметром 2 он будет равен $[1, 1, 1, -1, -1, 2, -1, 2]$. Первая порядковая статистика этого массива равна -1.

8. Восьмой, девятый и десятый запросы требуют найти значения $F_{3,2}(3)$, $F_{3,2}(4)$ и $F_{3,2}(8)$, то есть третью, четвертую и восьмую порядковые статистики в массиве $[1, 1, 1, -1, -1, 2, -1, 2]$. Они равны соответственно -1, 1 и 2.

Система оценки

Тесты к этой задаче состоят из двенадцати групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих 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. Итоговый балл за каждую группу равняется максимальному баллу, полученному за эту группу тестов по всем отправленным посылкам.

Если в подзадаче указаны ограничения на m или k, то они распространяются как на m_0 и k_0, так и на параметры всех запросов первого и третьего типов.

Группа	Баллы	Доп. ограничения				Необх. группы	Комментарий
		n	m	k	q		
0	0	–	–	–	–	–	Тесты из условия.
1	4	$n \leqslant 1000$	$m \leqslant 1000$	–	$q = 0$	–	–
2	5	–	–	$k = 1$	$q = 0$	–	–
3	6	–	–	$k = 1$	$q \leqslant 100\,000$	2	$t_j = 1$ для всех запросов
4	7	–	–	$k = 1$	$q \leqslant 100\,000$	2, 3	$t_j \neq 3$ для всех запросов
5	11	–	–	$k = 2$	$q = 0$	–	–
6	9	–	$m \leqslant 10^6$	–	$q = 0$	1	–
7	10	$n \leqslant 1000$	–	–	$q = 0$	1	–
8	7	–	–	–	$q = 0$	1, 2, 5 – 7	–
9	11	–	–	–	$q \leqslant 100\,000$	1 – 3, 5 – 8	$t_j = 1$ для всех запросов
10	13	–	–	–	$q \leqslant 100\,000$	1 – 3, 5 – 9	$t_j \neq 2$ для всех запросов
11	9	–	–	–	$q \leqslant 100\,000$	0 – 10	–
12	8	–	–	–	–	0 – 11	**Offline-проверка.**