

Задача А. Эрен и подвал

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

Эрен хочет открыть подвал своего дома, но, к сожалению, потерял ключ. Альтернативный способ попасть в подвал — ввести на замке разблокирующий его код доступа.

Замок на двери подвала устроен следующим образом:

- на нем есть два кодовых механизма, первый из которых изначально указывает на число a , а второй — на число b ;
- первый кодовый механизм сломан, поэтому изменить значение a нельзя;
- второй кодовый механизм можно вращать только в одном направлении, тем самым увеличивая значение b ;
- замок открывается тогда и только тогда, когда существует целое число $d > 1$, делящее и a , и b (иными словами, когда у a и b есть общий делитель больше единицы).

За одну секунду Эрен может повернуть второй кодовый механизм так, что b увеличится ровно на 1. Определите, за какое минимальное время Эрен сможет открыть подвал.

Формат входных данных

В первой строке дано единственное целое число t — количество тестовых случаев ($1 \leq t \leq 1000$).

В i -й из следующих t строк через пробел даны два целых числа a_i и b_i — начальные значения, на которые указывают кодовые механизмы в i -м тесте ($2 \leq a_i, b_i \leq 10^9$).

Формат выходных данных

Для каждого теста выведите в отдельной строке минимальное время, за которое Эрен откроет подвал.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

Все указанные ниже дополнительные ограничения выполняются для всех тестовых случаев тестов группы.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$t \leq 10, a_i \leq 1000, b_i \leq 100$		полная
2	23	$t \leq 100, a_i, b_i \leq 10000$	1	полная
3	25	a_i — простое число		полная
4	42	нет	1 – 3	первая ошибка

Пример

стандартный ввод	стандартный вывод
3	2
15 7	12
23 11	0
35 42	

Задача В. Необычная сортировка

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

Магане любит брать обычные вещи и выворачивать их смысл наизнанку. В этот раз ей попался совершенно обычный массив целых чисел, и чтобы избавиться от скуки, она хочет сделать его отсортированным по неубыванию.

Для этого Магане может один раз выбрать произвольный набор различных позиций в массиве и заменить элементы на этих позициях на противоположные, то есть умножить их на -1 . Например, чтобы сделать массив $[-4, 4, 1, 3, -10]$ отсортированным, она может умножить на -1 числа на позициях 2 и 5, и получить массив $[-4, -4, 1, 3, 10]$.

К сожалению, надолго ее такое занятие не увлечет, поэтому Магане решила посчитать количество способов отсортировать массив по неубыванию указанным образом. Два способа считаются различными, если различаются выбранные ей наборы позиций.

Помогите ей с этой задачей! Поскольку итоговое количество способов может быть слишком большим, найдите ответ по модулю 998244353.

Формат входных данных

В первой строке ввода записано целое число n — количество элементов в массиве ($1 \leq n \leq 10^6$).

Во второй строке через пробел перечислены n целых чисел a_1, a_2, \dots, a_n — элементы массива ($-10^9 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — количество способов отсортировать массив указанным образом (по модулю 998244353).

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	7	$n \leq 3$	–	полная
2	12	$n \leq 10$	1	полная
3	9	$1 \leq a_i \leq n$, все элементы массива различны	–	полная
4	19	$1 \leq a_i$, все элементы массива различны	3	полная
5	13	существует i , при котором $a_i = 0$	–	полная
6	18	каждое число либо встречается в массиве дважды, либо не встречается вообще	–	полная
7	22	нет	1 – 6	первая ошибка

Примеры

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

Задача С. Колода для фокусов

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

Недавно были выпущены колоды игральных карт нового формата. Значение каждой карты в колоде задается `base64`-строкой длины k . В рамках данной задачи под `base64`-строкой подразумевается строка, каждый символ которой может принимать одно из следующих 64 значений: ‘#’, ‘\$’, ‘0’–‘9’, ‘A’–‘Z’ и ‘a’–‘z’.

Каждая колода состоит из $n = 64^k$ карт и содержит ровно по одной карте с каждым возможным значением. Более того, в запечатанной новой колоде карты расположены в порядке лексикографического возрастания их значений. Напомним, что строка s лексикографически больше строки t тогда и только тогда, когда для некоторого i верно, что $s_1 = t_1, \dots, s_{i-1} = t_{i-1}$, а $s_i > t_i$. Иными словами, когда после их общего (возможно, пустого) префикса в s идет больший символ, чем в t .

Фокусник Чоно Урёку перемешивает колоду, бесконечное количество раз повторяя следующее действие:

1. сначала он разделяет колоду на две половины — в первой оказываются карты, находившиеся на позициях от 1 до $\frac{n}{2}$, во второй — от $\frac{n}{2} + 1$ до n (порядок карт не меняется);
2. затем он собирает колоду заново, строго чередуя карты из двух половин, начиная с первой: если до перемешивания пронумеровать карты от 1 до n , то после перемешивания колода будет выглядеть следующим образом:

$$\langle 1, \frac{n}{2} + 1, 2, \frac{n}{2} + 2, \dots, \frac{n}{2}, n \rangle.$$

Чоно собирается показать m фокусов. Чтобы i -й фокус удался, ему необходимо, чтобы карта со значением x_i оказалась на позиции, на которой в изначальной (запечатанной) колоде находилась карта со значением y_i . Считая, что для каждого фокуса Чоно берет новую запечатанную колоду, помогите ему определить, какое минимальное число перемешиваний понадобится для каждого фокуса.

Формат входных данных

В первой строке ввода через пробел даны два целых числа k и m — длина значения каждой карты и количество фокусов ($1 \leq k \leq 2500$; $1 \leq m \leq 1000$).

В следующих m строках перечислены описания фокусов. В i -й их них через пробел даны две строки x_i и y_i — значения карт, фигурирующих в i -м фокусе ($|x_i| = |y_i| = k$). Гарантируется, что x_i и y_i состоят только из указанных в условии символов.

Формат выходных данных

Для каждого фокуса выведите в отдельной строке номер перемешивания, после которого карта со значением x_i впервые окажется на позиции y_i .

Если карта изначально находится на нужной позиции, выведите 0. Если же такое событие никогда не произойдет, выведите -1 .

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	8	$k = 1, m \leq 200$	–	полная
2	11	$k \leq 10$, все x_i равны между собой	–	полная
3	13	$k \leq 2$	1	полная
4	12	$k \leq 50, m \leq 200$	–	полная
5	18	$k, m \leq 300$	4	полная
6	19	$k = 1024, m \leq 700$	–	первая ошибка
7	19	нет	1 – 6	первая ошибка

Примеры

стандартный ввод	стандартный вывод
1 7	0
# #	1
U \$	-1
4 3	3
t D	3
D t	4
2 \$	2
\$ 2	
3 8	2
Hd7 CYZ	13
mZ\$ 1Z8	15
iYq poa	-1
JlP edh	13
SyR uxw	17
aCp n50	-1
I#g 0q8	1
wrP tir	

Замечание

ASCII коды используемых в значениях карт символов принимают значения 35 ('#'), 36 ('\$'), от 48 до 57 (цифры), от 65 до 90 (заглавные латинские буквы) и от 97 до 122 (строчные латинские буквы).

Задача D. Испытание силомера

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

Сайтама выполняет последовательные удары по силомеру. Силомер представляет из себя массив целых чисел длины n . Изначально i -е число массива равно a_i для всех i .

Вам необходимо обработать q событий, происходящих с силомером. Событие номер i может быть одного из трех типов:

1. подходит наблюдатель и просит посчитать сумму чисел массива на отрезке $[l_i; r_i]$, то есть величину $a_{l_i} + a_{l_i+1} + \dots + a_{r_i}$;
2. Сайтама наносит обычный удар силы x_i по отрезку $[l_i; r_i]$: всем элементам массива на позициях от l_i до r_i включительно присваивается значение x_i
3. Сайтама наносит сильный удар по отрезку $[l_i; r_i]$: для всех j от l_i до r_i включительно происходит присваивание $a_j \leftarrow \text{popcount}(a_j)$.

Здесь $\text{popcount}(x)$ — это количество единичных бит в двоичной записи числа x . Иными словами, при событии третьего типа каждое число на отрезке события заменяется на количество своих единичных бит.

На каждый подход наблюдателя, то есть событие первого типа, сообщите ему интересующую его сумму.

Формат входных данных

В первой строке записаны два целых числа n и q — длина массива и количество событий ($1 \leq n, q \leq 2 \cdot 10^5$).

Во второй строке через пробел записаны n целых чисел a_1, \dots, a_n — начальные элементы массива силомера ($0 \leq a_i \leq 10^9$).

Следующие q строк описывают события. Первое число t_i в описании события — тип события ($1 \leq t \leq 3$). Следующие два заданные через пробел числа — это границы отрезка l_i и r_i ($1 \leq l_i \leq r_i \leq n$). Если это событие второго типа, то есть $t_i = 2$, далее следует число x_i , обозначающее, что надо выполнить присваивания $a_j \leftarrow x_i$ для всех $l_i \leq j \leq r_i$ ($0 \leq x_i \leq 10^9$).

Формат выходных данных

Для каждого события первого типа выведите в отдельной строке сумму элементов массива на отрезке, заданном этим событием.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	12	$n, q \leq 5000$	–	полная
2	7	все a_i и x_i — степени двойки	–	полная
3	4	все a_i и x_i — степени двойки или нули	2	полная
4	9	$t_i \neq 2$ для всех i	–	полная
5	5	для всех событий второго типа $l_i = r_i$	4	полная
6	12	$a_i, x_i \leq 20$	–	полная
7	15	$n \leq 10^5$ и $q \leq 10^4$	1	полная
8	16	все события первого типа следуют строго позже событий второго и третьего типов	–	полная
9	20	нет	1 – 8	первая ошибка

Примеры

стандартный ввод	стандартный вывод
6 6 9 1 3 1 6 3 2 2 3 9 2 4 5 10 3 2 4 1 2 4 1 2 6 3 2 5	6 19
5 6 0 3 4 1 1 1 2 5 2 2 3 8 3 1 4 3 2 2 1 3 4 2 3 5 5	9 2

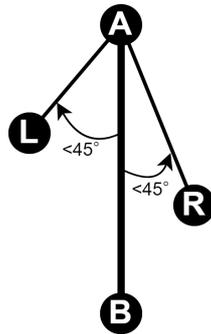
Задача Е. Гигантский дракон

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Канеки смотрит на неориентированный граф на плоскости из n вершин и m ребер. В этом графе ему интересно найти самого большого дракона.

Назовем *сегментом* дракона три ребра графа AL , AB и AR , имеющие общую вершину A , и обладающие следующими свойствами:

- $0 < \angle(BAL) < 45^\circ$ и направление поворота от \vec{AB} к \vec{AL} — по часовой стрелке;
- $0 < \angle(BAR) < 45^\circ$ и направление поворота от \vec{AB} к \vec{AR} — против часовой стрелки;
- $|AB| \geq |AL|$ и $|AB| \geq |AR|$, то есть AB — максимальное по длине из трех ребер.



При выполнении всех указанных условий вершины A и B называются *началом* и *концом* сегмента, а ребра AL , AB и AR — *левой лапой*, *основанием* и *правой лапой* сегмента, соответственно.

Определим *дракона* как последовательность сегментов, в которой

- начало первого сегмента A_1 , также называемое *головой* дракона, находится в вершине S ;
- $A_i = B_{i-1}$ для всех $i > 1$, то есть начало каждого следующего сегмента совпадает с концом предыдущего;
- $\left| \angle \left(\vec{A_{i-1}B_{i-1}}, \vec{A_iB_i} \right) \right| < 45^\circ$, то есть угол между векторами оснований соседних сегментов строго меньше 45° ;
- $\left| \angle \left(\vec{A_1A_i}, \vec{A_iB_i} \right) \right| < 45^\circ$, то есть угол между вектором от головы дракона A_1 до начала сегмента и основанием сегмента строго меньше 45° .

Обратите внимание, что здесь углы взяты по модулю, то есть каждый следующий сегмент может быть повернут относительно предыдущего на менее чем 45° как по, так и против часовой стрелки.

Мощностью дракона будем считать сумму **квадратов длин** оснований его сегментов, то есть $\sum |A_iB_i|^2$. В заданном графе помогите Канеки найти дракона максимальной мощности с головой в вершине S .

Формат входных данных

В первой строке входных данных даны три числа n, m, S ($2 \leq n \leq 2 \cdot 10^5$; $1 \leq m \leq 4 \cdot 10^5$; $1 \leq S \leq n$) — количество вершин и ребер в заданном графе и номер вершины, являющейся головой дракона.

В следующих n строках дано описание вершин графа. Каждая строка содержит два целых числа x_i и y_i — координаты i -й вершины ($0 \leq x_i, y_i \leq 10^9$). Гарантируется, что все вершины графа различны, то есть не существует двух вершин, обе координаты которых совпадают.

Далее следует пустая строка.

В следующих m строках дано описание ребер графа. Каждая строка содержит два целых числа u_i и v_i — номера вершин, соединенных i -м ребром ($1 \leq u_i, v_i \leq n$; $u_i \neq v_i$). Гарантируется, что граф не содержит кратных ребер.

Формат выходных данных

В первой строке выходных данных выведите два числа k и ans — количество сегментов в драконе, имеющем максимальную мощность, и само значение его мощности.

В следующих k строках выведите описание сегментов в том порядке, в котором они образуют дракона. В качестве описания сегмента i выведите номера вершин L_i , B_i и R_i .

Будем считать, что дракон может состоять только из вершины S . В таком случае количество сегментов и его мощность следует считать нулями.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи и необходимых подзадач, а также тесты из условия успешно пройдены.

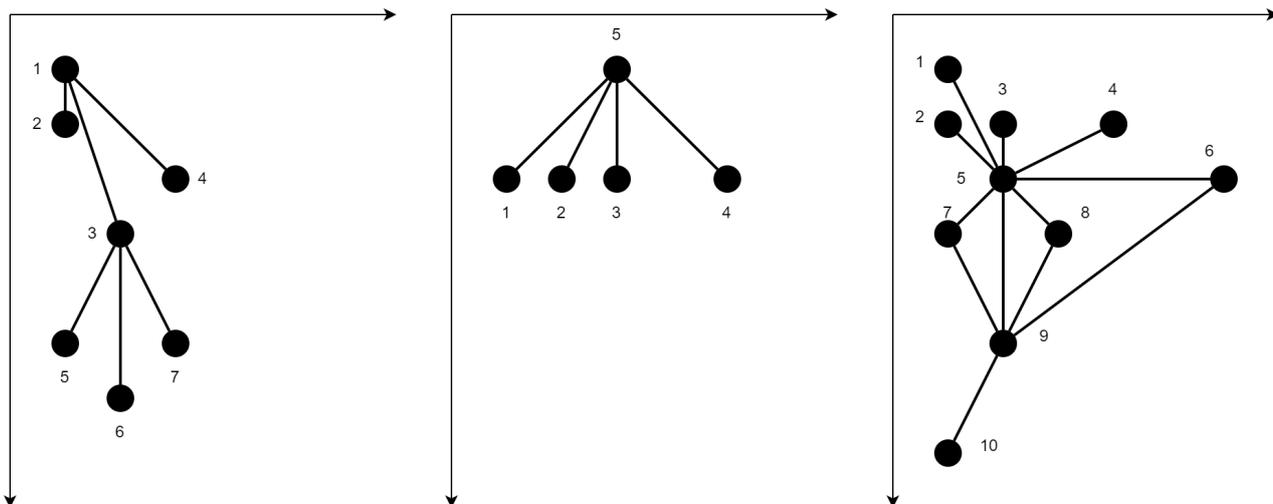
Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	12	$n, m \leq 5$		полная
2	18	$n, m \leq 10$	1	полная
3	16	граф — дерево		полная
4	16	$0 \leq x_i \leq 1$ для всех i		полная
5	18	$n \leq 10^3, m \leq 10^4$	2	первая ошибка
6	20	нет	1 – 5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
7 6 1 1 1 1 2 2 4 3 3 1 6 2 7 3 6 1 2 1 3 1 4 3 5 3 6 3 7	2 19 2 3 4 5 6 7
5 4 5 1 3 2 3 3 3 5 3 3 1 1 5 2 5 5 3 4 5	0 0
10 12 9 1 1 1 2 2 2 4 2 2 3 6 3 1 4 3 4 2 6 1 8 1 5 2 5 3 5 4 5 5 6 5 9 6 9 7 9 8 9 9 10 5 8 5 7	2 14 8 5 7 3 1 2

Замечание

Графы, данные в первом, втором и третьем тесте условий, выглядят следующим образом.



- В первом тесте в качестве максимального дракона можно взять весь граф целиком;
- Во втором тесте ни одна тройка ребер не может быть взята в сегмент, так как не выполняется одно из обязательных условий;
- В третьем тесте максимальный дракон состоит из двух сегментов с основаниями $9 \rightarrow 5$ и $5 \rightarrow 1$ с лапами $(9 \rightarrow 8, 9 \rightarrow 7)$ и $(5 \rightarrow 3, 5 \rightarrow 2)$.