

Задача А. Оптимизация Матрицы

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

Искусственный интеллект, поддерживающий работу Матрицы — сложная система, поэтому Архитектор решил попробовать оптимизировать ее.

Эту систему можно представить в виде n последовательно соединенных узлов. Каждый узел обладает специальной характеристикой w_i — *вычислительной важностью*.

Архитектор хочет выбрать некоторый узел и распространить его действие на k соседних узлов в одном и в другом направлении. Если в каком-то из направлений узлов меньше чем k , то он распространит его действие на столько узлов, сколько есть. Будем считать, что он распространил действие выбранного узла суммарно на d узлов. Тогда вычислительная важность выбранного узла станет равна $(1 + d) \cdot w_i$, а вычислительная важность узлов, на которое распространилось его действие, будет равна 0.

Помогите Архитектору Матрицы понять, какую максимальную суммарную вычислительную важность системы можно получить, выполнив указанное действие.

Формат входных данных

В первой строке входных данных дается два целых числа n и k — количество вычислительных узлов и то, на сколько узлов распространяется действие выбранного узла с одной стороны ($1 \leq n \leq 2 \cdot 10^5$; $0 \leq k \leq 2 \cdot 10^5$).

Во второй строке через пробел следуют n положительных чисел $w_1, w_2, \dots, w_n$ ($1 \leq w_i \leq 5 \cdot 10^5$) — вычислительная важность каждого из узлов.

Формат выходных данных

В первой и единственной строке выходных данных выведите одно целое число — максимальную вычислительную важность системы, которую можно получить.

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	10	$n \leq 3$		первая ошибка
2	18	$n \leq 6$	1	первая ошибка
3	20	$k = 0$		первая ошибка
4	28	$k \leq 1000$	2	первая ошибка
5	24	нет	1 – 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 1 1 3 2	9
5 1 1 5 3 2 4	21

Задача В. Нужно больше энергии

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 1024 мегабайта

Машины матрицы потребляют невообразимое количество энергии. Некоторые люди, взаимодействуя в Матрице, выделяют очень много энергии, поэтому Архитектор Матрицы задумался над новым помещением для расположения крио-камер с людьми.

По задумке, новое помещение будет представлять собой прямоугольную комнату длины n и ширины x , вдоль горизонтальной стороны которой будет расположена большая аккумуляторная батарея для накопления энергии. В каждом из рядов помещения (для $1 \leq i \leq n$) планируется разместить одну криокамеру на некотором расстоянии от нижней стены p_i ($1 \leq p_i \leq x$).

Криокамера расположена **энергетически выгодно**, если криокамеры слева и справа (если такие есть) расположены на меньшем расстоянии от аккумулятора, чем текущая криокамера, то есть $p_{i-1} < p_i$ и $p_i > p_{i+1}$.

Назовем конфигурацией криокамер набор чисел $p_1, \dots, p_n$. Архитектор Матрицы задумался, сколько существует конфигураций расположения криокамер в комнате длины n , таких что энергетически выгодно расположенных криокамер будет ровно k . Он смог ответить на этот вопрос, а сможете ли ответить на него вы?

Так как ответ на задачу может быть достаточно большим, найдите его по модулю числа $10^9 + 7$.

Формат входных данных

В первой и единственной строке ввода через пробел даны три целых числа n , x и k — длина комнаты, максимальное расстояние, на котором может быть расположена криокамера от аккумулятора, и количество криокамер, расположенных энергетически выгодно ($1 \leq n, x, k \leq 500$).

Формат выходных данных

В единственной строке выведите ответ на задачу по модулю $10^9 + 7$ — количество конфигураций криокамер, удовлетворяющих условиям.

Система оценки

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

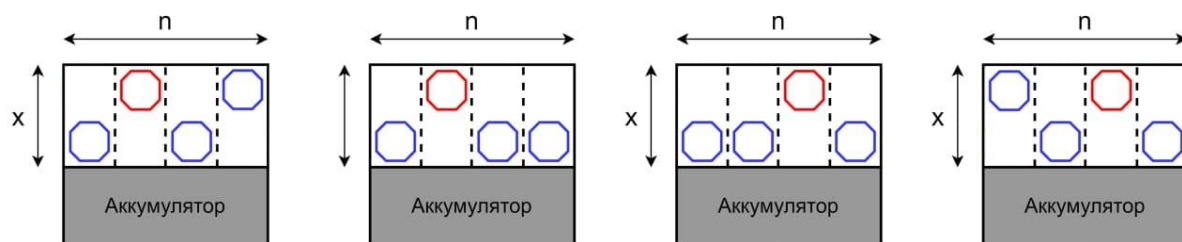
Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	20	$n, k, x \leq 10$		полная
2	35	$n, k, x \leq 50$	1	полная
3	15	$n, k, x \leq 300$	2	полная
4	10	$n, k, x \leq 400$	3	первая ошибка
5	20	нет	4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 2 1	4
10 5 4	351596

Замечание

Ниже находится иллюстрация к первому примеру:



Задача С. Финальное противостояние

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

Нео и Тринити противостоят армии ботов, заполонивших виртуальный город Сан-Франциско.

Сам город можно представить в виде бесконечной координатной прямой. Боты нападают последовательными волнами, каждая из n волн представляет собой отрезок этой прямой, который целиком заполнен ботами. Нео знает, что в i -ю волну боты будут занимать отрезок с концами в точках l_i и r_i (то есть $[l_i, r_i]$).

Тринити считает группу последовательных волн с a -й по b -ю включительно *опасной*, если при пересечении занятых ботами в эти волны отрезков имеет длину не меньше m_1 и не больше m_2 . Иными словами, пара чисел (a, b) , для которых $a \leq b$, задает опасную группу волн тогда и только тогда, когда

$$m_1 \leq \left| \bigcap_{a \leq i \leq b} [l_i, r_i] \right| \leq m_2.$$

Например, если $n = 3$ и $[l_1, r_1] = [4, 7]$, $[l_2, r_2] = [5, 8]$ и $[l_3, r_3] = [3, 6]$, то пересечение отрезков первых двух волн равно $[4, 7] \cap [5, 8] = [5, 7]$ и имеет длину 2, а пересечение всех трех отрезков равно $[5, 6]$ и имеет длину 1. При $m_1 = 2$, первые две волны вместе будут опасными, а все три вместе — нет. Обратите внимание, что если какая-то группа волн является опасной, это не значит, что любая содержащая ее группа волн также опасна.

Помогите Нео и Тринити найти **количество** опасных групп волн, чтобы они могли поскорее выбрать план действий.

Формат входных данных

В первой строке ввода через пробел даны три целых числа n , m_1 и m_2 — количество волн и границы допустимых значений длины пересечения отрезков ($1 \leq n \leq 2 \cdot 10^5$; $0 \leq m_1 \leq m_2 \leq 10^9$).

В следующих n строках перечислены координаты концов отрезков, заполненных ботами: в i -й строке через пробел даны целые числа l_i и r_i — границы i -го отрезка ($0 \leq l_i < r_i \leq 10^9$).

Формат выходных данных

Выведите единственное целое число — количество различных групп из последовательных волн, пересечение занятых ботами отрезков в которых имеет длину между m_1 и m_2 .

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	12	$n \leq 100$		полная
2	18	$l_i \leq l_{i+1}$ для всех i		полная
3	18	$n \leq 1000$	1	первая ошибка
4	20	$r_i - l_i \leq 30$ для всех i		полная
5	32	нет	1 – 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
4 1 2 0 4 1 5 2 6 3 4	5
6 0 1 0 4 0 2 0 1 3 4 2 4 0 4	15

Замечание

В первом примере в качестве опасной группы подойдет любая группа, содержащая четвертую волну, так как ее длина равна 1 и ее отрезок содержится во всех остальных отрезках, а значит и пересечение будет иметь длину 1. А также, помимо этого, подойдет группа из первых трех волн, имеющая пересечение отрезков, равное $[2, 4]$.

Во втором примере подойдет любая группа, содержащая третью или четвертую волну. Несложно заметить, что группы, состоящие только из двух первых или двух последних волн, имеют пересечение отрезков длины хотя бы 2. Всего таких подходящих групп ровно 15.

Задача D. Распределенная Матрица

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

В условиях нехватки энергии Матрица была модифицирована, чтобы расходовать как можно меньше энергетических ресурсов. Всего есть n узлов Матрицы с уникальными номерами от 1 до n . Изначально энергия есть только в генераторе, который является узлом с номером 1.

Требующие энергии узлы постепенно подключаются к уже запитанным узлам, и начинают получать энергию от них, образуя сеть питания в виде дерева. Некоторые узлы могут отказывать и восстанавливаться спустя время после отказа. Уже подключенный к сети узел никогда не переподключается к другим узлам, даже если какой-то из косвенно питающих его узлов отказал.

Ненадежностью узла называется время, прошедшее с момента его последнего восстановления (либо с момента подключения этого узла к сети, если он еще не отказывал). Временем подключения генератора считается момент времени 0.

Требуется обработать m событий. Событие номер i происходит ровно спустя i секунд от начала и может быть одного из следующих видов:

1. «! $x_i y_i$ » — узел y_i подключается к узлу x_i и начинает получать энергию от него;
2. «- x_i » — узел x_i отказывает и перестает проводить энергию;
3. «+ x_i » — ранее отказавший узел x_i восстанавливается и продолжает проводить энергию;
4. «? $x_i y_i$ » — требуется выяснить, насколько надежна пара узлов x_i и y_i .

Для ответа на запрос последнего типа требуется проверить, получают ли энергию оба узла x_i и y_i . Узел получает энергию, если сам подключен к сети, и все узлы на пути от генератора до него включительно находятся в исправном состоянии (не отказали). Если оба узла x_i и y_i получают энергию, требуется вывести суммарную ненадежность всех узлов, от которых зависит работа хотя бы одного из узлов x_i или y_i (то есть узлов, расположенных на путях от них до генератора).

Формат входных данных

В первой строке ввода через пробел даны два целых числа n и m — общее количество узлов, которым требуется питание, и количество событий, которые надо обработать ($2 \leq n, m \leq 2 \cdot 10^5$).

В i -й из следующих m строк дано описание i -го запроса (который происходит в момент времени i). Описание формата запросов дано в условии. За символом, обозначающим тип запроса, в зависимости от этого типа, следует либо одно целое число x_i , либо два целых числа x_i и y_i , разделенные пробелом — номера задействованных в запросе узлов ($1 \leq x_i, y_i \leq n$; $x_i \neq y_i$).

Гарантируется, что в запросе первого типа узел x_i уже подключен к сети, а y_i — нет. Также для запросов второго и третьего типа гарантируется, что узел x_i отказывает только если был до этого исправен, и наоборот, восстанавливается только после соответствующего отказа.

Формат выходных данных

После каждого запроса четвертого типа следует в отдельной строке вывести ответ на этот запрос. Если хотя бы один из узлов x_i и y_i не получает энергию, следует вывести «-1» (без кавычек). Если же оба узла получают энергию, следует вывести целое число, равное сумме ненадежностей всех узлов, лежащих на путях от генератора до x_i и y_i .

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	12	$n, m \leq 1000$, нет запросов второго и третьего типов (узлы не отказывают)		полная
2	14	$n, m \leq 1000$	1	полная
3	18	нет запросов второго и третьего типов (узлы не отказывают)	1	полная
4	24	нет запросов третьего типа (отказавшие узлы не восстанавливаются)	3	полная
5	32	нет	1 – 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 7 ! 1 2 ! 1 3 ? 2 3 - 3 ? 2 3 + 3 ? 2 3	6 -1 14
3 7 ! 1 2 ! 2 3 ? 1 3 - 2 ? 1 3 + 2 ? 1 3	6 -1 13

Замечание

В первом примере отказ третьего узла, очевидно, влечет ответ «-1» на второй запрос «? 2 3». Ответ на первый запрос равен 6, так как с момента подключения к сети генератора, второго узла и третьего, прошли 3, 2 и 1 секунда, соответственно. В момент третьего запроса с момента подключения генератора и второго узла прошло 7 и 6 секунд, соответственно, тогда как третий узел вернулся в строй ровно 1 секунду назад, что дает ответ 14.

Во втором примере отказ второго узла аналогично влечет ответ «-1» на второй запрос. Ответ на первый запрос вычисляется так же, как и в первом примере, а на третий — как $7 + 1 + 5 = 13$.