

Задача А. Очередь в магазине

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

В одном известном магазине случилась распродажа, однако администрация не учла одну проблему: в магазине всего одна касса! Сразу после начала распродажи возле кассы организовалась длинная очередь. Никто не любит очереди, поэтому у покупателей постепенно возрастает уровень агрессии. От вас требуется рассмотреть процесс продвижения очереди.

Могут происходить события трёх типов:

1. В конец очереди встал человек с уровнем агрессии a ;
2. Первый человек в очереди начал ругаться с кассиром, в результате чего уровень его агрессии увеличился на x , а уровень агрессии каждого из **остальных** людей в очереди (если в очереди стоит не один человек) увеличился на y ;
3. Первый человек в очереди оплатил покупку и ушёл из магазина.

От вас требуется обработать N событий. Будем считать, что изначально очередь пуста. Так как администрация магазина заботится о своей репутации, им важно знать, насколько агрессивными их покупатели уходят из магазина. Поэтому для каждого события третьего типа нужно определить уровень агрессии человека, который ушёл из магазина.

Формат входных данных

В первой строке записано одно число N — количество событий ($2 \leq N \leq 300000$).

В каждой из следующих N строк содержится описание очередного события:

- 1 a , если произошло событие первого типа;
- 2 x y , если произошло событие второго типа;
- 3, если произошло событие третьего типа.

Для всех событий верно, что $1 \leq a, x, y \leq 10^9$. Гарантируется, что события второго и третьего типов происходят только в том случае, если в очереди есть хотя бы один человек. Также гарантируется, что после N событий в очереди не останется ни одного человека. Возможны случаи, когда первый человек в очереди несколько раз подряд ссорится с кассиром.

Формат выходных данных

Для каждого запроса третьего типа выведите одно число — уровень агрессии человека, который ушёл из магазина. Каждое число следует выводить на отдельной строке.

Система оценки

Подзадача	Баллы	Дополнительные ограничения	Оценка	Необходимые подзадачи
0	0	Тесты из условия	подзадача	—
1	40	$2 \leq N \leq 1000$ Для всех событий $1 \leq a, x, y \leq 1000$	подзадача	—
2	60	Дополнительных ограничений нет	подзадача	1

Пример

стандартный ввод	стандартный вывод
8	10
1 4	13
1 2	1
2 6 1	
3	
2 10 20	
1 1	
3	
3	

Замечание

Сначала в очередь встали два человека с уровнями агрессии 4 и 2 соответственно. Затем первый человек поссорился с кассиром, после чего уровни агрессии людей стали равны 10 и 3. После этого первый человек ушёл из очереди, а второй поссорился с кассиром. Теперь уровень его агрессии равен 13. Затем в очередь встал человек с уровнем агрессии 1, после чего оба человека ушли из магазина.

Задача В. Скоростной транспорт

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

На прямой трассе проводятся испытания двух видов скоростного транспорта: поезда на магнитной подушке маглева, и вакуумного поезда гиперлуп. Трасса поделена на участки длиной 1 километр, для испытаний маглева выделен трек, состоящий из участков с a -го по b -й, а для испытаний гиперлупа — с c -го по d -й. Испытательные треки не имеют общих участков, сначала следует трек для испытаний маглева, а затем трек для испытаний гиперлупа. Таким образом, выполнены неравенства $a \leq b < c \leq d$.

По итогам первой фазы испытаний выяснилось, что длину трека для гиперлупа желательно увеличить, в то же время длину трека для маглева можно уменьшить. Экспериментаторы планируют выбрать новые параметры треков для второй фазы испытаний. Необходимо выбрать значения A , B , C и D и использовать на второй фазе для испытаний маглева трек, состоящий из участков с A -го по B -й, а для испытаний гиперлупа трек, состоящий из участков с C -го по D -й.

При этом для сокращения расходов на переоборудование треков необходимо выполнение следующих условий:

Условие	Формула
Новый трек для маглева полностью содержится в старом треке	$a \leq A \leq B \leq b$
Новый трек для гиперлупа полностью содержит старый трек	$C \leq c \leq d \leq D$
Суммарная длина треков не должна измениться	$(d - c + 1) + (b - a + 1) = (D - C + 1) + (B - A + 1)$
Новые треки не должны иметь общих участков	$A \leq B < C \leq D$

Прежде чем выбрать новые параметры треков, учёным необходимо выяснить, сколько существует способов выбрать искомые значения A , B , C и D .

Требуется написать программу, которая по заданным значениям a , b , c и d определит количество способов выбрать искомые значения A , B , C и D .

Формат входных данных

На вход подаются четыре целых числа, по одному на строке: a , b , c и d ($1 \leq a \leq b < c \leq d \leq 10^5$).

Формат выходных данных

Выведите одно целое число: количество способов выбрать искомые значения A , B , C и D .

Система оценки

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

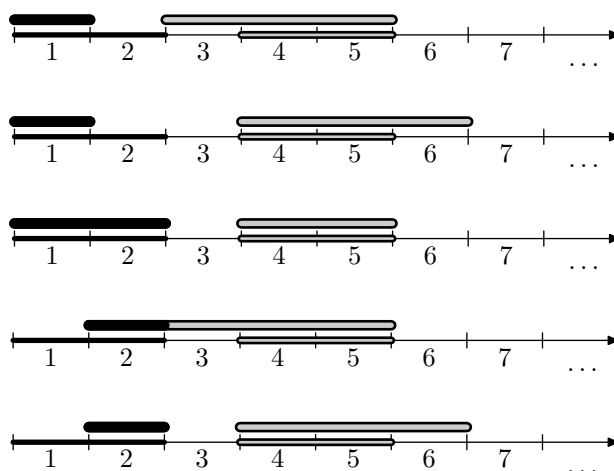
Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	21	$1 \leq a \leq b < c \leq d \leq 20$		полная
2	22	$1 \leq a \leq b < c \leq d \leq 500$	1	полная
3	28	$1 \leq a \leq b < c \leq d \leq 2000$	1, 2	полная
4	29	$1 \leq a \leq b < c \leq d \leq 10^5$	1, 2, 3	полная

Примеры

стандартный ввод	стандартный вывод
2 4 6 8	13
1 2 4 5	5

Замечание

Пять вариантов нового расположения треков во втором примере приведены на рисунке. На координатной оси отрезками показаны старые треки, над координатной осью жирными отрезками показаны новые треки.



Задача С. Продукты в экспедиции

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Ученые планируют набор продуктов для экспедиции на Марс. Планируется, что запас экспедиции будет состоять из n типов продуктов, пронумерованных целыми числами от 1 до n . У экспедиции будет k_i порций продуктов i -го типа. Продукт i -го типа должен быть использован на протяжении t_i дней после начала экспедиции, после чего портится. Если за t_i дней не все порции продукты i -го типа съедены, то все оставшиеся порции этого продукта уничтожаются.

В экспедицию планируют направить s участников. Каждый день участники экспедиции выбирают любые s имеющихся у них порций и съедают их. Разные участники экспедиции могут есть как одинаковые, так и различные типы продуктов.

Отдел планирования снабжения хочет понять, насколько избыточен набор продуктов, запланированный для экспедиции. Они хотят выяснить, какое максимальное различное количество типов продуктов участники экспедиции смогут полностью съесть в процессе экспедиции, не допустив уничтожения ни одной их порции продукта этого типа.

Требуется написать программу, которая по описанию продуктов и количеству участников экспедиции определяет максимальное количество типов продуктов, которые могут быть полностью съедены в процессе экспедиции.

Формат входных данных

В первой строке два целых числа n и s — количество типов продуктов и количество участников экспедиции ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq s \leq 10^9$).

В следующих n строках находится по два целых числа t_i , k_i — время, за которое портятся продукты i -го типа, и количество порций продукта i -го типа ($1 \leq t_i \leq 10^9$, $1 \leq k_i \leq 10^{18}$).

Формат выходных данных

Сначала выведите единственное целое число s ($0 \leq s \leq n$) — максимальное количество типов продуктов, которые могут быть полностью съедены в процессе экспедиции. В следующей строке выведите s целых чисел $p_1, p_2, \dots, p_s$ ($1 \leq p_i \leq n$, все p_i различны) — номера типов продуктов.

Если существует несколько подходящих множеств типов продуктов максимального размера, выведите любое из них. Типы продуктов можно выводить в любом порядке.

Система оценки

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

Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	5	$n = 1, 1 \leq c, t_i \leq 10^9, 1 \leq k_i \leq 10^{18}$		полные
2	22	$1 \leq n \leq 16, 1 \leq c, t_i \leq 10^9, 1 \leq k_i \leq 10^{18}$	1	полные
3	15	$1 \leq n \leq 2000, c = 1, 1 \leq t_i \leq 2000, 1 \leq k_i \leq 10^{18}$		первая ошибка
4	18	$1 \leq n \leq 2000, 1 \leq c, t_i \leq 10^9, 1 \leq k_i \leq 10^{18}$	1, 2, 3	первая ошибка
5	15	$1 \leq n \leq 2 \cdot 10^5, 1 \leq c, t_i \leq 10^9, 1 \leq k_i \leq 10^{18},$ все t_i совпадают	1	первая ошибка
6	25	$1 \leq n \leq 2 \cdot 10^5, 1 \leq c, t_i \leq 10^9, 1 \leq k_i \leq 10^{18}$	1 – 5	первая ошибка

Примеры

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

Задача D. Доставка почты

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

Для автоматизированной доставки почты в отдалённых регионах тестируется роботизированная система с использованием автоматизированного робота-курьера.

В регионе расположены n городов, соединённые $n - 1$ двусторонними дорогами. По дорогам можно добраться от каждого города до любого другого. Если два города соединены дорогой, назовём их соседними. У каждого города не более D соседних городов. Города пронумерованы от 1 до n , город номер 1 является столицей региона.

В каждом городе находится офис курьерской компании. Курьеру необходимо развезти m посылок. Для i -й посылки заданы два числа: a_i и b_i — город, из которого надо доставить посылку, и город, в который надо доставить посылку.

Робот-курьер действует по следующему алгоритму.

- Он начинает свой путь в столице и перемещается между городами по дорогам. При перемещении между городами курьер может перевозить на себе произвольное число посылок.
- Каждый раз, когда курьер приезжает в город, в котором он ранее не был, он заезжает в офис, находящийся в этом городе, оставляет все имеющиеся у него посылки, адресованные в этот город, и забирает все посылки, отправляемые из этого города.
- Если есть соседний с текущим город, в котором курьер ещё не был, он выбирает один из этих городов и перемещается в него.
- Если все соседние города посещены, и курьер находится не в столице, он перемещается в соседний город, ближайший к столице. Если курьер находится в столице, он прекращает свою поездку.

Обратите внимание, что курьер заезжает в офис в каждом городе ровно один раз, при первом посещении. Курьер доставит i -ю посылку, если он заедет в офис в a_i -м городе раньше, чем в офис в b_i -м городе.

Порядок, в котором курьер посещает города, может быть различным, в зависимости от выбора соседнего непосещённого города. Будем называть последовательность посещения городов *допустимой*, если все посылки будут доставлены.

Требуется написать программу, которая по заданному описанию дорог в регионе и посылок, которые необходимо доставить, определяет количество различных допустимых последовательностей посещения городов и выводит остаток от деления этого количества на $10^9 + 7$.

Формат входных данных

В первой строке даны два целых числа n и m — количество городов в регионе и количество посылок ($2 \leq n \leq 100\,000$, $1 \leq m \leq 300\,000$).

В следующих $n - 1$ строке даны описания дорог. Каждая дорога описывается двумя целыми числами u_i и v_i — номера городов, которые соединяет i -я дорога ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$).

Гарантируется, что по дорогам можно добраться от любого города до любого другого. Гарантируется, что количество дорог, выходящих из каждого города, не превосходит D .

В следующих m строках даны описания посылок. Каждая посылка описывается двумя целыми числами a_i и b_i — номера городов, из которого и в который нужно доставить i -ю посылку ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$).

Формат выходных данных

В единственной строке выведите одно целое число — ответ по модулю $10^9 + 7$.

Система оценки

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

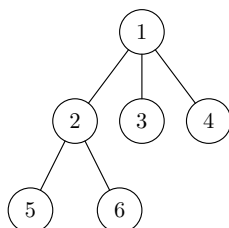
Подзадача	Баллы	Ограничения	Необходимые подзадачи	Информация о проверке
1	27	$n \leq 10, m \leq 50$		первая ошибка
2	9	$n \leq 20, m \leq 100, D \leq 3$		первая ошибка
3	7	$n \leq 1\,000, m \leq 1\,000, D \leq 3$	2	первая ошибка
4	8	$n \leq 1\,000, m \leq 1\,000, D \leq 7$	2, 3	баллы
5	6	$n \leq 1\,000, m \leq 1\,000, D \leq 12$	1, 2, 3, 4	баллы
6	12	$n \leq 100\,000, m \leq 300\,000, D \leq 3$	2, 3	первая ошибка
7	18	$n \leq 100\,000, m \leq 300\,000, D \leq 7$	2, 3, 4, 6	баллы
8	13	$n \leq 100\,000, m \leq 300\,000, D \leq 12$	1 – 7	баллы

Примеры

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

Замечание

Рассмотрим схему городов и дорог из первого примера, она приведена на рисунке. Необходимо доставить посылки из города 5 в город 3 и из города 4 в город 6.



Следующие последовательности посещения городов являются допустимыми. Первое посещение, во время которого курьер посещает офис, отмечено жирным: **(1, 4, 1, 2, 5, 2, 6, 2, 1, 3, 1)**, **(1, 4, 1, 2, 6, 2, 5, 2, 1, 3, 1)**.

Во втором примере для той же схемы городов и дорог необходимо доставить посылку из города 5 в город 2. Это невозможно, так как при повторном посещении города 2 курьер не заезжает в офис, а впервые посетить город 5 до города 2 при пути из столицы невозможно.