

Заключительный этап

Задача по искусственному интеллекту

Предскажи пол

Условия:

VK продвигает сервисы, товары и услуги разнообразными способами. Одним из основных является реклама. Качественная реклама требует хорошего понимания аудитории. Вам предложен датасет. Необходимо предсказать пол человека.

Представьте, что вы работаете в бизнес-юните VK Реклама компании VK.

Приближаются праздники. Крупная компания, занимающаяся продажей подарочных наборов, хочет запустить рекламу подарков в соцсетях ВКонтакте, Одноклассники и Дзен.

Вам нужно разработать модель машинного обучения, которая будет предсказывать пол пользователя соцсетей. Это необходимо, чтобы реклама была эффективной: демонстрировалась пользователям соответствующего гендера.

Компания VK заинтересована в получении этого крупного заказа. Сотрудник, который представит модель машинного обучения, быстро и точно предсказывающую пол пользователя, получит вознаграждение.

В вашем распоряжении есть табличный датасет, на котором вы можете тренировать модели машинного обучения. У вас есть 4 часа и 10 попыток, чтобы разработать свой вариант модели машинного обучения.

Описание датасета

Таблицы

train_labels.csv:

- «user_id» - id пользователя;
- «target» - пол пользователя (1 / 0).

train.csv, test.csv;

- «request_ts» - server timestamp of request;
- «user_id» - id пользователя (см. п.1);
- «referer» - url, где показывается реклама. В данном случае зашифровано 2 части url:
 - 1) domain - домен urlа;
 - 2) path - все что после domain. Например, <https://a758bf6/1432d3f1>, a 758bf6 - domain, 1432d3f1 - path.
- «geo_id» - id geo;
- «user_agent» - строка user_agent.

referer_vectors.csv:

- «component0» - ... - «component9» - числа, которые несут в себе информацию о url. Их нельзя как-либо интерпретировать;
- «referer» - url, где показывается реклама (см. п.2).

geo_info.csv:

- «geo_id» - id geo (см. п.2);
- «country_id» - id страны;

- «region_id» - id региона;
- «timezone» - часовой пояс для geo.

Требования

Для каждого пользователя (user) из файла test_users.csv необходимо предсказать пол. Их рекламные запросы лежат в файле test.csv.

Формат вывода

Формат вывода соответствует train_labels.csv.

Задача A. AI Angry Birds

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 0.5 секунд

Ограничение по памяти: 256 мегабайт

ИИ Маруся учится играть в одну из версий игры Angry Birds. Напоминаем, что суть игры заключается в том, чтобы, стреляя из рогатки птичками, уничтожить сооружения, в которых находятся зелёные свинки. В этой версии игры есть 4 домика, в которых находятся зелёные свинки. Крепкость i -го домика равна a_i .

Маруся может стрелять только одним видом птичек. Птичка этого вида имеет силу g , то есть попадание в i -й домик (а Маруся стреляет без промахов) приводит к понижению его крепкости на g . Если крепкость домика после попадания перестала быть положительной, то домик разрушен и находившиеся в нём свинки скрылись в неизвестном направлении. В противном случае домик ещё стоит и до следующего выстрела Маруси свинки из неразрушенных домиков, за исключением свинок из i -го домика, успеют частично починить i -й домик, а именно свинки из j -го домика, если тот ещё не разрушен, повысят крепкость i -го домика на b_j . Например, если на данный момент разрушен только первый домик, а во второй домик Маруся только что стреляла, то до следующего выстрела крепкость второго домика повысится на b_3+b_4 . Заметьте, что по итогу описанного процесса крепкость домика может даже стать больше изначальной.

Приведите тактику для разрушения всех домиков или сообщите, что разрушить все домики невозможно. Заметьте, что не требуется минимизировать суммарное количество выстрелов, но запрещается стрелять в уже разрушенный домик.

Формат входных данных

В i -й из первых четырёх строк ввода содержится два целых числа a_i и b_i , описывающих i -й домик ($0 \leq a_i, b_i \leq 10^{18}$). Если a_i положительно, то i -й домик имеет крепкость a_i . Если a_i равно 0, то домик уже разрушен.

В последней, пятой строке содержится число g – сила птички ($1 \leq g \leq 10^{18}$).

Обратите внимание, что входные данные и ответ могут быть достаточно большими, поэтому следует использовать 64-битный тип данных, например `long long` в C/C++, `long` в Java, `int64` в Pascal.

Формат выходных данных

Очевидно, что не имеет смысла переключаться с одной цели на другую до тех пор, пока изначальная цель не уничтожена. Поэтому в качестве ответа, если он существует, выведите 4 строки. i -я строка должна содержать два целых числа k_i и c_i — номер домика, который будет разрушен i -м, и количество выстрелов из рогатки по k_i -му домику.

Если домик был разрушен изначально, то его нужно вывести тоже, причём не важно в какой из строк.

Если правильных ответов несколько, то выведите любой их них. Стрелять в уже разрушенный домик нельзя и заметьте, что не требуется минимизировать суммарное количество выстрелов. Если ответа не существует, выведите одно число -1 .

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
0	–	примеры из условия		полная
1	10	$a_2 = a_3 = a_4 = 0$ $a_i, b_i, g \leq 10^9$ для всех i		первая ошибка
2	15	$a_3 = a_4 = 0$ $a_i, b_i, g \leq 10^9$ для всех i	1	первая ошибка
3	15	$a_4 = 0$ $a_i, b_i, g \leq 10^9$ для всех i	0, 1, 2	первая ошибка
4	15	$b_i = 0$ для всех i		первая ошибка
5	20	$a_i, b_i, g \leq 10^6$ для всех i		первая ошибка
6	10	$a_i, b_i, g \leq 10^9$ для всех i	0 – 3, 5	первая ошибка
7	15	без дополнительных ограничений	0 – 6	первая ошибка

Пример

стандартный ввод	стандартный вывод
6 3	1 2
3 1	3 2
7 2	4 0
0 70	2 1
5	

Пояснение к примеру

После первого выстрела в первый домик его крепкость стала равна $6 - 5 = 1$. После этого свинки из второго и третьего домиков починили первый домик и его крепкость стала равна $1 + 1 + 2 = 4$. После этого первый домик был разрушен вторым выстрелом.

После первого выстрела в третий домик его крепкость стала равна $7 - 5 = 2$. После этого свинки из второго домика починили третий домик и его крепкость стала равна $2 + 1 = 3$. После этого третий домик был разрушен следующим выстрелом, а следующим выстрелом был разрушен и второй домик.

Существуют и другие правильные ответы, поэтому вывод вашей программы может отличаться от указанного в условии задачи.

Задача В. Перестановка с условиями

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Артем хочет найти перестановку длины n , в которой a элементов больше обоих своих соседей и b элементов меньше обоих своих соседей.

Перестановка — это последовательность длины n целых чисел от 1 до n , в которой все числа встречаются ровно по одному разу. Например, (1) , $(4, 3, 5, 1, 2)$, $(3, 2, 1)$ — перестановки, а $(1, 1)$, $(4, 3, 1)$, $(2, 3, 4)$ — нет.

Формат входных данных

В первой строке содержится три целых числа n , a и b ($3 \leq n \leq 10^5$; $2 \leq a, b \leq 10^6$) — длина перестановки, число элементов больше обоих своих соседей и число элементов меньше обоих своих соседей.

Формат выходных данных

В качестве ответа выведите «YES» (без кавычек), если существует перестановка, и «NO» в противном случае. Если перестановка существует, то выведите её на второй строке.

Слова «YES» и «NO» можно выводить в любом регистре, например, «YES», «Yes», «yEs», и так далее.

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
0	–	примеры из условия		полная
1	10	$n \leq 5$		первая ошибка
2	15	$n \leq 10$	1	первая ошибка
3	20	$a < b$		первая ошибка
4	20	$a > b$		первая ошибка
5	15	$a = b$		первая ошибка
6	20	без дополнительных ограничений	1–5	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 2 2	No
7 1 0	Yes 1 7 6 5 4 3 2
7 3 2	Yes 1 3 2 5 4 7 6

Задача С. Капли дождя...

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

В детстве Артем любил наблюдать за капельками на запотевшем окне. Иногда эти капельки устраивали очень увлекательные гонки!

Сейчас Артем уже не маленький мальчик, чтобы просто любоваться этими гонками. Сейчас он уже способен сказать, какая капелька придет к «финишу» первой, зная некоторую информацию об этих капельках.

А способны ли вы ответить на этот вопрос?

Формально процесс выглядит следующим образом: вам дана бесконечная плоскость (запотевшее окно), на этой плоскости располагается n точек (капельки воды).

Для каждой капельки известна следующая информация — x , y , $size$ координаты капельки на окне и размер капельки.

Каждая капелька, в силу своей природы, старается течь вниз по стеклу. В нашей интерпретации мы будем говорить, что каждая капелька стремится уменьшить свою y координату до величины, равной 0.

За каждую единицу пройденного расстояния размер капельки может уменьшиться на d (это происходит в следствие того, что капелька оставляет за собой след на стекле).

Размер капельки уменьшается, если она движется не по чужому следу, а прокладывает свой. Формально изменение, происходящее с положением капельки за секунду, можно выразить следующей схемой:

1. Если капелька достигла координаты $y = 0$, то она не двигается.
2. Капелька пытается изменить свою y координату на 1, не теряя в размере.
Это возможно, если существовала некоторая другая капелька, которая уже прошла этим путем.
3. Капелька пытается изменить свою y координату на 1, оставляя за собой след, и уменьшается в размере на d .
Это возможно, если размер капельки не меньше, чем d .
4. Если капелька не смогла переместиться согласно условиям пунктов выше, то она не двигается.

В том случае, если при перемещении капельки она оказывается в позиции, где есть другая капелька, тогда эти капельки сливаются, и их размеры складываются.

Для лучшего осознания процесса обратитесь к пояснениям в примере.

Артем попросил вас посчитать, через какое минимальное время найдется капелька, которая достигнет «финиша» — координаты $y = 0$. Если это никогда не произойдет — выведите -1 .

Формат входных данных

В первой строке вводится два числа $1 \leq n, d \leq 10^5$ — изначальное количество капелек на плоскости и величина, на которую каждая капелька уменьшается при изменении своей y координаты.

В следующих n строках вводится информация про n капелек воды.

В i строке вводится три числа $1 \leq x, y, size \leq 10^9$ — позиция i капли в начальный момент времени и изначальный размер капли.

Гарантируется, что все пары x_i, y_j различны.

Формат выходных данных

Выведите минимальное время, через которое какая-нибудь из капелек достигнет координаты 0 или -1, если это никогда не произойдет.

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
0	–	примеры из условия		полная
1	15	$n \leq 100, x_i = 1$		первая ошибка
2	20	$n \leq 1000$	1	первая ошибка
3	30	$MAX(y_i) \leq 10^6$. Все $size_i$ – уникальны		первая ошибка
4	35	без дополнительных ограничений	1–3	первая ошибка

Примеры

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

Замечание

Пояснение к примеру 3.

Нулевой момент времени:

- Первая капля имеет координаты $(1, 2)$, а её размер равен 3
- Вторая капля имеет координаты $(1, 4)$, а её размер равен 5

Первый момент времени:

- Первая капля уменьшает свою y координату на 1 и уменьшается в размере на 2.
- Первая капля имеет координаты $(1, 1)$, а её размер равен 1.
- Вторая капля уменьшает свою y координату на 1 и уменьшается в размере на 2.
- Вторая капля имеет координаты $(1, 3)$, а её размер равен 3

Второй момент времени:

- Первая капля не может уменьшить свою y координату на 1, вследствие недостаточности размера.
- Первая капля имеет координаты $(1, 1)$, а её размер равен 1.
- Вторая капля уменьшает свою y координату на 1 и уменьшается в размере на 2.
- Вторая капля имеет координаты $(1, 2)$, а её размер равен 1

Третий момент времени:

- Первая капля не может уменьшить свою y координату на 1, вследствие недостаточности размера.
- Первая капля имеет координаты $(1, 1)$, а её размер равен 1.
- Вторая капля уменьшает свою y координату на 1, но не уменьшается в размере, ведь в момент времени 1 первая капля уже совершала такое изменение и оставляла свой след.
- Вторая капля попадает в координату с первой и их размеры сливаются.
- Объединенная капля имеет координаты $(1, 1)$, а её размер равен 2.

Четвертый момент времени:

- Объединенная капля уменьшает свою y координату на 1 и уменьшается в размере на 2.
- Объединенная капля имеет координаты $(1, 0)$, а её размер равен 0.
- Объединенная капля достигла финиша в момент времени 4.

Задача D. Прыгающий робот

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Дано натуральное число n ($1 \leq n \leq 2 \cdot 10^5$). Также дано два массива размера n : a_1, \dots, a_n ($1 \leq a_i \leq 10^9$) и b_1, \dots, b_n ($1 \leq b_i \leq 10^9$).

У вас есть робот, который может прыгать по массиву a . А именно, находясь на i -м элементе, он может прыгнуть вправо на ближайший элемент j ($i < j$) такой, что $a_j > a_i$. При этом, находясь на i -м элементе массива, робот должен заплатить b_i монет. Если он не может заплатить за очередной элемент в этой последовательности, робот не может совершить прыжок, и процесс останавливается.

Вы хотите q раз сделать следующее: поставить робота на позицию s_i ($1 \leq s_i \leq n$) и дать ему m_i ($1 \leq m_i \leq 10^{18}$) монет, после чего посмотреть, сколько прыжков он сможет сделать.

Ответьте на этот вопрос для разных значений s_i и m_i .

Формат входных данных

В первой строке задано натуральное число n ($1 \leq n \leq 2 \cdot 10^5$).

Во второй строке через пробел записан массив a_1, \dots, a_n ($1 \leq a_i \leq 10^9$). В третьей строке через пробел записан массив b_1, \dots, b_n ($1 \leq b_i \leq 10^9$). В четвертой строке записано натуральное число q ($1 \leq q \leq 2 \cdot 10^5$).

В следующих q строках через пробел записано по два числа: в i -й строке записаны числа s_i ($1 \leq s_i \leq n$) и m_i ($1 \leq m_i \leq 10^{18}$).

Формат выходных данных

Выведите q строк: в i -й строке должно быть записано единственное число — ответ на i -й запрос.

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
0	–	примеры из условия		полная
1	10	$n, q \leq 1000$	–	первая ошибка
2	20	$a_i < a_{i+1}$ для всех $0 \leq i \leq n - 1$	–	первая ошибка
3	30	$n \leq 5000$	1	первая ошибка
4	40	без дополнительных ограничений	1–3	первая ошибка

Пример

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

Задача E. Плохое слово и путь в матрице

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 1024 мегабайта

Вы просто проснулись в теле хомяка, в левой верхней клетке матрицы размера $n \times m$, зная лишь, что получить обратно ваше имя и вашу память вы сможете только решив задачу ниже.

Хомяк хочет попасть из стартовой клетки — $(1, 1)$, в клетку правого нижнего угла (n, m) .

Хомяк может переходить в любую смежную с ним клетку. Клетки считаются смежными, если они имеют общую сторону.

Введем понятие строка пути — это такая строка, которую мы получим, если выпишем символы во всех клетках, в которых бывал хомяк, двигаясь по некоторому пути. Отметим, что символы нужно выписывать в том порядке, в котором хомяк посещал их, двигаясь по конкретному пути. Назовем путь хорошим, если его строка пути не содержит плохое слово s как подстроку.

Вас просят найти кратчайший хороший путь из клетки $(1, 1)$ в клетку (n, m) , для заданного плохого слова s .

Формат входных данных

В первой строке входных данных находится три натуральных числа n, m, k — размеры матрицы и длина плохого слова s ($1 \leq n, m \leq 10^5, 1 \leq k \leq 10^2$).

В следующих n строках вводится матрица, состоящая из строчных букв латинского алфавита. В последней строке на вход подается плохое слово s , состоящее из строчных букв латинского алфавита.

Формат выходных данных

В единственной строке выведите ответ на задачу — минимальную длину хорошего пути. Если не существует хорошего пути — выведите -1 .

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
0	–	примеры из условия		полная
1	30	Все символы строки s одинаковы.		первая ошибка
2	30	$n \cdot m \leq 10^3, k \leq 10, s$ не содержит других символов кроме 'a', 'b'.		первая ошибка
3	40	без дополнительных ограничений	1–2	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 3 1 bbb aab bbb baa bbb a	10
1 1 1 f f	-1
1 8 4 bbbaabab bbaa	9