

Задача Minecraft. Задачка на подстроочки

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

Филипп очень любит задачки на строчечки. Он уже решил все известные ему задачки, но этого ему было мало. Поэтому Филипп решил придумать свою собственную задачку.

Для этого он взял строку t и набор из n строк $s_1, s_2, s_3, \dots, s_n$. У Филиппа есть m запросов, в i -м из них Филипп хочет взять подстроку строки t с l_i -го по r_i -й символ, и посчитать число её подстрок, которые совпадают с какой-то строкой из набора. Более формально, Филипп хочет посчитать число пар позиций a, b , таких что $l_i \leq a \leq b \leq r_i$, и подстрока строки t с a -го по b -й символ совпадает с некоторой строкой s_j из набора.

Подстрокой строки t с a -го по b -й символ называется строка, полученная из t путём удаления $a - 1$ символа из начала и $|t| - b$ символами из конца, где $|t|$ обозначает длину строки t .

Филипп уже решил эту задачу, а сможете ли вы?

Формат входных данных

В первой строке два целых положительных числа n и m ($1 \leq n, m \leq 500\,000$) — число строк в наборе и количество запросов.

Во второй строке дана единственная строка t , состоящая из строчных букв английского алфавита ($1 \leq |t| \leq 5 \cdot 10^6$).

В следующих n строках описываются строки из набора. В i -й из них дана единственная строка s_i , состоящая из строчных букв английского алфавита. Обозначим за S суммарную длину всех строк из набора. Гарантируется, что $S \leq 10^6$, а так же что все строки s_i различны.

В следующих строках вводятся запросы. В i -й из них даны два целых положительных числа l_i и r_i ($1 \leq l_i \leq r_i \leq |t|$) — левая и правая граница подстроки t из i -го запроса.

Формат выходных данных

В единственной строке выведите m целых чисел, i -е из них должно быть равно ответу на i -й запрос.

Примеры

ВВОД	ВЫВОД
3 5 abacaba aba a ac 1 7 1 3 2 7 2 5 4 5	7 3 5 3 1
4 4 abcdca ab ca bcd openolympiad 1 5 2 2 2 6 1 6	2 0 2 3

Пояснение

В первом примере в первом запросе требуется у всей строки посчитать число подстрок, которые входят в набор. Со строкой «aba» совпадают подстроки [1, 3] и [4, 6]. Со строкой «a» совпадают подстроки [1, 1], [3, 3], [5, 5], [7, 7]. Со строкой «ac» совпадает подстрока [3, 4]. Всего получается, что 7 подстрок строки «abasaba» совпадают со строками из набора.

Во втором запросе от исходной строки берется подстрока с 1 по 3 позицию, это строка «aba». В неё строка «aba» входит 1 раз, строка «a» входит 2 раза и строка «ac» не входит ни одного раза как подстрока.

В третьем запросе от исходной строки берется подстрока с 2 по 7 позицию, это строка «basaba». В неё строка «aba» входит 1 раз, строка «a» входит 3 раза и строка «ac» входит 1 раз как подстрока.

Система оценки

Тесты к этой задаче состоят из 9 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения				Необх. группы	Комментарий
		n	m	$ t $	S		
0	0	–	–	–	–	–	Тесты из условия.
1	10	$n \leq 100$	$m \leq 100$	$ t \leq 100$	$S \leq 10\,000$	0	
2	12	$n \leq 100$	$m \leq 500$	$ t \leq 5000$	–	0, 1	
3	7	$n \leq 5000$	–	$ t \leq 5000$	–	0, 1, 2	
4	8	$n \leq 100$	–	$ t \leq 50\,000$	–	0, 1, 2	
5	12	–	–	$ t \leq 100\,000$	$S \leq 100\,000$	0, 1	
6	8	–	–	$ t \leq 250\,000$	$S \leq 100\,000$	0, 1, 5	
7	7	–	–	$ t \leq 500\,000$	$S \leq 100\,000$	0, 1, 5, 6	
8	7	–	–	$ t \leq 750\,000$	$S \leq 100\,000$	0, 1, 5, 6, 7	
9	29	–	–	–	–	0 – 8	Offline-проверка.

Задача Dota. Покупка подарков

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

У маленького Саши есть две подруги, которых он хочет порадовать подарками на Восьмое марта. Для этого он отправился в самый большой торговый центр в городе.

В торговом центре есть n отделов, в каждом из которых находятся ровно два магазина. Для удобства пронумеруем отделы целыми числами от 1 до n . Известно, что подарки в первом магазине i -го отдела стоят a_i рублей, а во втором магазине i -го отдела — b_i рублей.

Войдя в торговый центр, Саша посетит каждый из n отделов торгового центра, причем в каждом отделе он зайдет ровно в один магазин. Таким образом, когда Саша попадет в i -й отдел, он выполнит ровно одно из двух действий:

1. Купить подарок первой подруге, потратив на это a_i рублей.
2. Купить подарок второй подруге, потратив на это b_i рублей.

Для каждой подруги Саша собирается купить хотя бы один подарок. Более того, он хочет подобрать подарки таким образом, чтобы разность цен самых дорогих подарков, купленных подругам, была как можно меньше, чтобы никто не обиделся.

Более формально: пусть m_1 — максимальная цена подарка, купленного первой подруге, а m_2 — максимальная цена подарка, купленного второй подруге. Саша хочет выбрать подарки таким образом, чтобы минимизировать величину $|m_1 - m_2|$.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 500\,000$) — количество отделов в торговом центре.

Каждая из следующих n строк содержит два целых числа a_i и b_i ($0 \leq a_i, b_i \leq 10^9$) — цены подарков в первом и втором магазине i -го отдела, соответственно.

Формат выходных данных

Выведите одно целое число — минимальную разность цен самых дорогих подарков, купленных подругам.

Примеры

ВВОД	ВЫВОД
2 1 2 2 1	0
5 1 5 2 7 3 3 4 10 2 5	1

Пояснение

В первом примере у Саши есть два возможных варианта действий: купить подарок первой подруге в первом отделе, а второй подруге — во втором отделе, или наоборот. В первом случае $m_1 = m_2 = 1$, а во втором случае — $m_1 = m_2 = 2$. В обоих случаях ответ равен 0.

Во втором примере можно купить подарки для первой подруги в 2-м, 4-м и 5-м отделах, а для второй подруги — в 1-м и 3-м отделах. Таким образом, $m_1 = \max(2, 4, 2) = 4$, $m_2 = \max(5, 3) = 5$. Ответ равен $|4 - 5| = 1$.

Система оценки

Тесты к этой задаче состоят из 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	a_i и b_i		
0	0	–	–	–	Тесты из условия.
1	16	$n \leq 20$	–	0	–
2	17	$n \leq 500$	–	0, 1	–
3	22	$n \leq 5000$	–	0, 1, 2	–
4	12	–	$a_i = b_i$	–	–
5	33	–	–	0 – 4	–

Задача GTA. Поиск фальшивых монет

Имя входного файла:	<code>input.txt</code> или стандартный поток ввода
Имя выходного файла:	<code>output.txt</code> или стандартный поток вывода
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Это интерактивная задача.

Перед вами партия из n золотых монет, среди которых есть k фальшивых. Все монеты выложены в ряд. Предполагаемый вес i -й монеты равен i грамм. Если монета фальшивая, ее вес равен 0 грамм.

Монеты трогать запрещено и единственная доступная вам операция — это выбрать некоторое $1 \leq p \leq n$ и взвесить первые p монет. В результате вам будет сказан настоящий суммарный вес этих монет.

Используя как можно меньше операций узнайте, какие k монет являются фальшивыми. Количество баллов будет зависеть от количество запросов, сделанных вашим решением, подробности смотрите в системе оценки.

Протокол взаимодействия

Каждый тест состоит из t игр, в которых вы должны узнать, какие монеты фальшивые. В самой первой строке находится единственное целое число t ($1 \leq t \leq 50$) — количество игр. Каждая игра состоит из взаимодействия в описанном ниже формате. После окончания всех игр, ваша программа должна завершиться.

В начале каждой игры вам дается два целых числа n и k ($1 \leq n \leq 10^9$, $1 \leq k \leq \min(100, n)$). После этого вы можете сделать несколько запросов взвешивания.

Для того чтобы сделать запрос взвешивания выведите «? p ». В результате вам будет возвращено единственное целое число a . Если $a = -1$, ваша программа превысила допустимый лимит запросов взвешивания на игру и должна немедленно завершиться. На каждую игру разрешено сделать не более 3500 запросов взвешивания. Иначе $a \geq 0$ это настоящий суммарный вес монет $1, 2, \dots, p$.

Для того чтобы сказать угаданное множество, выведите «! $i_1 i_2 \dots i_k$ », где $1 \leq i_1, i_2, \dots, i_k \leq n$ это различные индексы фальшивых монет в произвольном порядке. В результате вам будет возвращено единственное целое число a . Если $a = -1$, то ваш ответ неправильный и ваша программа должна немедленно завершиться. Иначе $a = 1$, и ваша программа должна продолжить взаимодействие следующей игрой или завершиться, если это была последняя игра.

Обратите внимание, что интерактор играет **адаптивно**. Не гарантируется, что множество фальшивых монет зафиксировано до начала игры. Единственное что гарантируется — в любой момент времени ответы, которые были даны в течение игры, соответствуют хотя бы одному множеству фальшивых монет. Ваш ответ на игру является правильным, если он соответствует всем ответам на запросы, данным в течение игры, а также не существует ни одного другого множества фальшивых монет, которое соответствует всем ответам.

После вывода каждого действия вашей программы выводите перевод строки. После вывода каждого действия вашей программы делайте сброс потока вывода.

Если вы используете «`writeln`» в Паскале, «`cout << ... << endl`» в C++, «`System.out.println`» в Java, «`print`» в Python, «`Console.WriteLine`» в C#, то сброс потока вывода у вас происходит автоматически, дополнительно ничего делать не требуется. Если вы используете другой способ вывода, рекомендуется делать сброс потока вывода. Обратите внимание, что перевод строки надо выводить в любом случае.

Пример

ВВОД	ВЫВОД
2	
3 2	? 3
2	! 1 3
1	
10 4	? 5
13	? 6
13	? 8
20	? 10
29	! 10 8 6 2
1	

Пояснение

В первой игре монеты 1, 3 являются фальшивыми. Таким образом, настоящие веса монет это $[0, 2, 0]$. С помощью одного запроса мы узнаем их суммарный вес 2, после чего однозначно можно восстановить множество фальшивых монет.

Во второй игре монеты 2, 6, 8, 10 являются фальшивыми. Таким образом, настоящие веса монет это $[1, 0, 3, 4, 5, 0, 7, 0, 9, 0]$. По ответам на запросы взвешивания можно однозначно восстановить множество фальшивых монет.

Система оценки

Тесты к этой задаче состоят из 6 групп. Пусть q это количество запросов взвешивания, которое сделало решение в ходе одной игры.

Баллы за первые 5 групп ставятся только при прохождении всех тестов группы и всех тестов необходимых групп. Для каждой из первых 5 групп зафиксировано некоторое число $maxQ$. Тест в первых 5 группах считается пройденным, если $q \leq maxQ$.

Балл за каждую игру в последней группе равняется $\min\left(50, \left\lfloor 50\sqrt{\frac{k+30}{q}} \right\rfloor\right)$. Общее количество баллов за тест равняется минимальному количеству баллов за все игры. Общее количество баллов за последнюю группу равняется минимальному количеству баллов за все тесты в этой группе.

Обратите внимание, что решение получает 100 баллов, если оно делает $\leq k + 30$ запросов взвешивания во всех тестах во всех играх.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		n	k	$maxQ$		
0	0	–	–	$maxQ = 3500$	–	Тесты из условия.
1	5	$n \leq 1000$	–	$maxQ = 1000$	0	
2	9	$n \leq 1000$	–	$maxQ = 600$	0, 1	
3	10	–	$k \leq 30$	$maxQ = 1000$	0	
4	13	–	$k = 3$	$maxQ = 33$	–	
5	13	–	$k = 4$	$maxQ = 34$	–	
6	≤ 50	–	–	$maxQ = 3500$	–	Частичные баллы.

Задача Counter-Strike. Путь домой

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

Известный фокусник Боря Будини путешествовал по стране X , которая состоит из n городов. Однако случилось несчастье, и его обокрали в городе номер 1. Теперь Будини предстоит нелегкий путь домой в город n .

Добираться он собирается самолетами. Всего в стране есть m авиарейсов, i -й летит из a_i в b_i и стоит s_i . Чтобы им воспользоваться, Боря должен быть в городе a_i и иметь на руках хотя бы s_i денег (которые он потратит на перелет).

После ограбления у него осталось всего p рублей, однако он не отчаивается! Находясь в городе i , он может хоть каждый день организовывать представления, которые будут приносить ему по w_i рублей.

Помогите фокуснику узнать, сможет ли он добраться до дома, а также какое минимальное количество представлений придется для этого организовать.

Формат входных данных

Первая строка содержит четыре целых числа n , m , p и g ($2 \leq n \leq 800$, $1 \leq m \leq 3000$, $0 \leq p \leq 10^9$, $0 \leq g \leq 6$) — количество городов, количество авиарейсов, изначальное количество рублей и номер группы тестов.

Во второй строке даны n целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^9$) — прибыль от представлений.

В следующих m строках даны по три целых числа a_i, b_i и s_i ($1 \leq a_i, b_i \leq n$, $1 \leq s_i \leq 10^9$) — начальный и конечный город, а также стоимость i -го авиарейса.

Формат выходных данных

Выведите единственное целое число — минимальное количество представлений, которое придется организовать Боре, чтобы добраться до дома, или -1 , если это сделать невозможно.

Примеры

ВВОД	ВЫВОД
4 4 2 0 7 4 3 1 1 2 21 3 2 6 1 3 8 2 4 11	4
4 4 10 0 1 2 10 1 1 2 20 2 4 30 1 3 25 3 4 89	24
4 4 7 0 5 1 6 2 1 2 5 2 3 10 3 4 50 3 4 70	10
4 1 2 0 1 1 1 1 1 3 2	-1

Пояснение

В первом примере Боре оптимально сделать 4 представления в первом городе, имея в итоге $2 + 7 \cdot 4 = 30$ рублей, а потом пройтись по маршруту $1 - 3 - 2 - 4$, потратив $6 + 8 + 11 = 25$ рублей.

Во втором примере Боре оптимально сделать 15 представлений в первом городе, полететь в 3 город, сделать там 9 представлений, и далее отправиться в 4 город.

Система оценки

Тесты к этой задаче состоят из 6 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, 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.

Группа	Баллы	Доп. ограничения				Необх. группы	Комментарий
		n	m	s_i	w_i		
0	0	–	–	–	–	–	Тесты из условия.
1	14	–	–	–	$w_i = 1$	–	
2	13	–	$m = n - 1$	–	–	–	$a_i = i, b_i = i + 1$
3	17	$n \leq 10$	–	–	–	0	
4	19	$n \leq 100$	–	$s_i \leq 100$	–	0	
5	21	$n \leq 100$	–	–	–	0, 3, 4	
6	16	–	–	–	–	0 – 5	Offline-проверка.