

Обзор задач

	Лучники	Прием на работу	POI	Измюм
Тип	Batch	Batch	Batch	Batch
Детальный ответ проверяющей системы	Частичный	Отсутствует	Полный	Частичный
Ограничение по времени (для каждого теста)	4 секунды	4 секунды	4 секунды	6 секунд
Ограничение по памяти (для каждого теста)	64 MB	64 MB	64 MB	128 MB
Баллы	100	100	100	100
	400			

Обзор задач

	Парковка Garage	Медведь Миша Mecho	Регионы Regions	Коммивояжер Salesman
Тип	Batch	Batch	Reactive	Batch
Детальный ответ проверяющей системы	Полный	Частичный	Отсутствует	Частичный
Ограничение по времени (для каждого теста)	1 секунда	2 секунды	12 секунд	5 секунд
Ограничение по памяти (для каждого теста)	32 MB	64 MB	128 MB	128 MB
Баллы	100	100	100	100
	400			

С++ программисты должны помнить, что использование С++ потоков (cin / cout) может привести к медленному вводу/выводу и, следовательно, к низкой производительности. Во избежание этого, пожалуйста, прочитайте рекомендации в технической памятке, расположенной по адресу <http://yeputons.net/tsweb/stats/09ioi/tech.pdf>

ЛУЧНИКИ

Соревнования лучников проходит по следующим правилам. Есть N мишеней, расположенных в линию и пронумерованных от 1 до N включительно в соответствии с их местом на линии (самая левая мишень имеет номер 1, а самая правая мишень имеет номер N). Также есть $2*N$ лучников. В произвольный момент во время турнира в каждую из мишеней стреляют два лучника. Каждый тур соревнований проходит по такой процедуре:

- два лучника, стреляющие в каждую мишень, соревнуются друг с другом и определяют Z:\победителя и проигравшего. После этого все лучники перемещаются следующим образом:
 - победители на мишенях с номерами от 2 до N включительно перемещаются на одну мишень левее (то есть, на мишени от 1 до $N - 1$ соответственно),
 - проигравшие на мишенях с номерами от 2 до N включительно, так же как победитель на мишени с номером 1, остаются на тех же мишенях,
 - проигравший на мишени с номером 1 перемещается на мишень с номером N .

Турнир состоит из R туров, количество туров не меньше, чем количество лучников (то есть $R \geq 2*N$).

Вы – единственный лучник, который прибыл на турнир точно вовремя. Остальные $2*N - 1$ лучников прибыли раньше и уже расположились в линию. Вам теперь нужно «вставиться» в линию где-то среди них. Вы знаете, что после того как вы займете свою позицию, два самых левых лучника начнут турнир на мишени с номером 1, два следующих на мишени с номером 2, и так далее. Два самых правых лучника начнут турнир на мишени с номером N .

Все $2*N$ лучников на турнире (включая вас) имеют ранг, установленный в соответствии с их навыками, при этом меньший ранг соответствует более высоким навыкам. Нет двух лучников с одинаковым рангом. Также, когда бы ни соревновались два лучника, всегда победит лучник с меньшим рангом.

Зная, как сильны в стрельбе ваши соперники, вы хотите разместиться так, чтобы закончить турнир на мишени с наименьшим возможным номером. Если есть несколько способов сделать это, выберите тот, который начинается на мишени с наибольшим номером.

ЗАДАНИЕ

Напишите программу, которая по заданным рангам всех лучников, включая вас, а также по расположению ваших соперников на линии, определяет, на какой мишени вы должны начать турнир, чтобы достичь описанных выше целей.

ОГРАНИЧЕНИЯ

$$1 \leq N \leq 200,000$$

Количество мишеней (также равно половине количества лучников).

$2*N \leq R \leq 1,000,000,000$ Количество туров турнира.
 $1 \leq S_k \leq 2*N$ Ранг лучника с номером k .

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать со стандартного потока ввода такие данные:

- Первая строка содержит целые числа N и R , разделенные пробелом.
- Последующие $2*N$ строк содержат ранги лучников. Первая из этих строк содержит ваш ранг. Остальные строки содержат ранги остальных лучников по одному в строке в том порядке, в котором они расположились в линии (слева направо). Каждая из этих $2*N$ строк содержит одно целое число из диапазона от 1 до $2*N$ включительно. Ранг 1 - наилучший, ранг $2*N$ - наихудший. Никакие два лучника не имеют одинакового ранга.

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна записать в стандартный поток вывода одну строку, в которой будет содержаться одно целое число между 1 и N включительно, являющееся номером мишени, на которой вы начнете турнир.

СИСТЕМА ОЦЕНКИ

Для набора тестов общей стоимостью 60 баллов N не будет превышать 5,000.

Также, для некоторых из этих тестов, суммарной стоимостью 20 баллов, N не будет превышать 200.

ПРИМЕРЫ

Пример ввода	Пример вывода
4 8 7 4 2 6 5 8 1 3	3

Вы второй наихудший лучник. Если вы начнете на мишени с номером 1, то вы попадете на мишень с номером 4 и останетесь там до конца. Если вы начнете на мишени с номером 2 или 4, вы просто останетесь там до конца турнира. Если вы начнете на мишени с номером 3, то победите наихудшего лучника, переместитесь на мишень с номером 2 и там останетесь.

Пример ввода	Пример вывода
4 9 2 1 5 8 3 4 7	2

6	
---	--

Вы второй наилучший лучник. Наилучший лучник уже расположен на мишени с номером 1, и останется там до окончания турнира. Таким образом, вне зависимости от того, где вы начнете турнир, вы будете всегда передвигаться со своей мишени, проходя через все мишени с номерами от 4 до 1 снова и снова. Для того чтобы окончить турнир на мишени с номером 1 после девяти переходов, вы должны начать на мишени с номером 2.

ПРИЁМ НА РАБОТУ

Вам необходимо нанять работников для строительного проекта. Заявление о приёме на работу подали N кандидатов, пронумерованных от 1 до N включительно. Каждый кандидат с номером k требует, чтобы в случае приёма его на работу ему платили не менее чем S_k долларов. Также для каждого кандидата с номером k известен его уровень квалификации Q_k . Положение о строительной деятельности требует, чтобы вы платили работникам пропорционально их уровню квалификации относительно друг друга. Например, если вы нанимаете двух работников A и B таких что $Q_A = 3 * Q_B$, то вы обязаны платить работнику A ровно в три раза больше, чем вы платите работнику B . Вам разрешается платить работникам нецелое число денег. Более того, разрешается даже платить количество денег, которое не может быть записано с помощью конечного числа десятичных цифр, такое как треть или шестую долю доллара.

В вашем распоряжении есть W долларов, и вы хотите нанять как можно больше рабочих. Вы решаете кого нанимать и сколько им платить, но вы должны удовлетворить как требованиям работников о минимальном жаловании, так и требованиям положения о строительной деятельности. Естественно, что вам требуется уложиться в бюджет, равный W долларам.

Для данного строительного проекта уровень квалификации работников не имеет значения. Вы заинтересованы только в том, чтобы нанять как можно больше работников независимо от их уровня квалификации. Однако, если есть несколько способов достичь цели, то вы хотите выбрать такой, чтобы общая сумма денег, которую вы заплатите работникам, была как можно меньше. Если и этого можно достичь несколькими способами, то нет никакого различия между этими способами, и вас удовлетворит любой из них.

ЗАДАНИЕ

Напишите программу, которая по заданным требованиям к жалованию и уровням квалификации кандидатов, а также количеству денег, которое у вас есть, определяет, каких кандидатов вам требуется нанять. Вы должны нанять как можно больше из них и при этом потратить как можно меньше денег, соблюдая требования положения о строительной деятельности, описанные выше.

ОГРАНИЧЕНИЯ

$1 \leq N \leq 500\,000$	Число кандидатов.
$1 \leq S_k \leq 20\,000$	Минимальное требование к жалованию кандидата номер k .
$1 \leq Q_k \leq 20\,000$	Уровень квалификации кандидата номер k .
$1 \leq W \leq 10\,000\,000\,000$	Сумма денег, доступная вам.

ВАЖНОЕ ЗАМЕЧАНИЕ

Максимальное значение W не может быть представлено 32-битным типом данных. Вам необходимо использовать 64-битный тип данных, такой как `long long` в C/C++ или `int64` в Pascal, чтобы значение W можно было сохранить в одной переменной. Дополнительные подробности представлены на страницах с технической информацией.

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать из стандартного потока ввода следующие данные:

- Первая строка входного файла содержит два целых числа N и W , разделённые пробелом.
- Следующие N строк описывают кандидатов, по одному кандидату на каждую строку. k -я строка из них описывает кандидата с номером k и содержит два целых числа S_k и Q_k , разделённых пробелом.

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна вывести в стандартный поток вывода следующие данные:

- Первая строка должна содержать одно целое число H – количество работников, которых вы принимаете на работу.
- Следующие H строк должны содержать список номеров кандидатов в произвольном порядке, которых вы выбрали для найма на работу (различные целые числа от 1 до N), по одному в каждой строке.

СИСТЕМА ОЦЕНКИ

Для каждого из тестов, используемых для проверки решения этой задачи, вы получаете полный балл, если ваш выбор кандидатов помогает достигнуть всех ваших целей при удовлетворении всем заданным ограничениям. Если вы выведете корректно первую строку (то есть, корректное значение H), но при этом оставшаяся часть файла не будет соответствовать вышеописанным условиям, то вы получите 50% баллов за этот тест. Это правило также действует даже в случае, если оставшаяся часть файла отформатирована неправильно, но первая строка выведена правильно.

Для набора тестов общей стоимостью 50 баллов значение N не будет превосходить 5 000.

ПРИМЕРЫ

Пример ввода	Пример вывода
4 100 5 1000 10 100 8 10 20 1	2 2 3

Единственная комбинация, при которой вы можете позволить себе нанять двух рабочих и удовлетворить всем требованиям – это выбрать рабочих с номерами 2 и 3. Вы можете заплатить им 80 и 8 долларов, соответственно, таким образом, уложившись в бюджет 100 долларов.

Пример ввода	Пример вывода
3 4 1 2 1 3 1 3	3 1 2 3

В этом примере вы можете позволить себе нанять всех трёх рабочих. Вы платите 1 доллар рабочему с номером 1 и по 1.50 доллара рабочим с номерами 2 и 3, и таким образом, укладываетесь в ваш бюджет, равный 4 долларам.

Пример ввода	Пример вывода
3 40 10 1 10 2 10 3	2 2 3

В этом примере вы не можете позволить себе нанять всех трёх рабочих, так как это стоило бы вам 60 долларов, но вы можете позволить себе нанять любых двух из них. Вы выбираете рабочих с номерами 2 и 3, потому что в этом случае получается наименьшая сумма денег по сравнению с другими комбинациями из двух рабочих. Вы можете заплатить 10 долларов рабочему с номером 2 и 15 долларов рабочему с номером 3, общая сумма будет равна 25 долларам. Если бы вы наняли рабочих с номерами 1 и 2, то вам пришлось бы заплатить им хотя бы 10 и 20 долларов соответственно. Если же бы вы выбрали рабочих с номерами 1 и 3, то вам пришлось бы заплатить им хотя бы 10 и 30 долларов соответственно.

POI

Full Feedback Problem

Пловдивская Олимпиада по информатике (POI) проходила согласно таким необычным правилам. Было N участников и T задач. Каждая задача оценивалась с использованием всего лишь одного теста. Таким образом, для каждого участника и каждой задачи было только две возможности: либо участник решил задачу, либо не решил ее. Не было частичного оценивания решения никакой задачи.

Количество баллов, назначенное каждой задаче, определялось после соревнования, и было равно количеству участников, которые не решили задачу. Баллы каждого участника подсчитывались как сумма баллов, назначенных задачам, решенным этим участником.

Филипп участвовал в соревновании, но он запутался в сложных правилах оценивания, и сейчас он, глядя на результаты, не в состоянии определить свое место в финальном протоколе. Помогите Филиппу написать программу, которая подсчитает его баллы и его место в финальном протоколе.

Перед соревнованием участникам присвоили уникальные номера от 1 до N включительно. Номер Филиппа обозначим P . В финальном протоколе участники перечислены в порядке убывания набранных ими баллов. В случае равенства баллов, первыми будут перечислены участники, которые решили больше задач. В случае равенства количества решенных задач, участники с одинаковыми результатами будут перечислены в порядке возрастания их номеров.

ЗАДАНИЕ

Напишите программу, которая по заданной информации о том, какие задачи были решены какими участниками, определит количество баллов у Филиппа и его место в финальном протоколе.

ОГРАНИЧЕНИЯ

$1 \leq N \leq 2,000$	Количество участников.
$1 \leq T \leq 2,000$	Количество задач.
$1 \leq P \leq N$	Номер Филиппа.

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна прочитать со стандартного потока ввода такие данные:

- Первая строка содержит целые числа N , T и P , разделенные пробелами.
- Последующие N строк описывают, какие задачи были решены какими участниками. k -ая строка из них описывает, какие задачи были решены участником с номером k . Каждая такая строка содержит T целых чисел, разделенных пробелом. Первое из этих чисел обозначает, решил ли первую задачу участник с номером k . Второе число обозначает то же для второй задачи, и так далее. Эти T чисел могут быть только 0 или 1, где 1 означает, что участник с номером k решил соответствующую задачу, и 0 означает, что он ее не решил.

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна записать в стандартный поток вывода одну строку с двумя целым числами, разделенными одним пробелом. Первое число – количество баллов, которые Филипп получил на соревновании POI. Второе число – место Филиппа в финальном протоколе. Место - это целое число от 1 до ***N*** включительно, где 1 означает, что участник расположен вверху финального протокола (то есть имеет наибольшее количество баллов), а ***N*** означает, что он расположен внизу финального протокола (то есть имеет наименьшее количество баллов).

СИСТЕМА ОЦЕНКИ

Для набора тестов общей стоимостью 35 баллов, никто из участников не наберет столько же баллов, как Филипп.

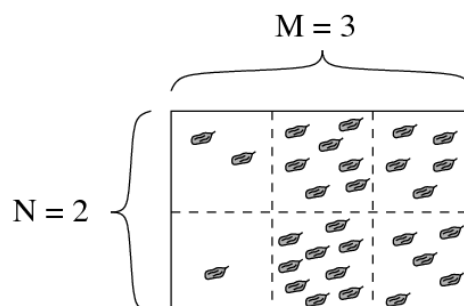
ПРИМЕР

ПРИМЕР ВВОДА	ПРИМЕР ВЫВОДА
5 3 2 0 0 1 1 1 0 1 0 0 1 1 0 1 1 0	3 2

Первая задача не была решена одним участником, то есть, она оценивается в 1 балл. Вторая задача не была решена двумя участниками, она оценивается в 2 балла. Третья задача не была решена четырьмя участниками, она оценивается в 4 балла. Таким образом, первый участник наберет 4 балла. Второй участник (Филипп), а также четвертый и пятый участники наберут по 3 балла каждый. Третий участник наберет 1 балл. Участник с номерами 2, 4 и 5 имеют одинаковое количество баллов и решили одно и то же количество задач, поэтому в соответствии со вторым правилом (в этом случае участники располагаются в порядке возрастания их номеров) Филипп окажется перед участниками с номерами 4 и 5. Таким образом, в финальном протоколе Филипп будет на втором месте, после участника с номером 1.

ИЗЮМ

Известная Пловдивская шоколадница Бонни хочет разрезать плитку шоколада с изюмом. Плитка имеет прямоугольную форму и состоит из одиночных квадратных кусочков. Кусочки выровнены параллельно краям плитки таким образом, что они формируют N строк и M столбцов, всего получается $N \cdot M$ кусочков. На каждом из кусочков имеется одна или более изюминок, и никакая изюминка не лежит между кусочками и не пересекает разрез между ними.



Изначально плитка шоколада представляет собой единое целое. Бонни хочет разрезать её на все меньшие и меньшие части, пока она, наконец, не разрежет всю плитку шоколада на $N \cdot M$ одиночных кусочков. Поскольку Бонни очень занята, она попросила своего ассистента Петра помочь разрезать плитку шоколада. Пётр делает только прямые разрезы от края до края части. Он хочет, чтобы ему платили за каждый разрез, который он сделает. У Бонни совсем нет денег, но у неё осталось бесконечное количество изюминок, и она собирается расплатиться с Петром изюмом. Петра это устраивает, но при следующем условии: каждый раз, когда он разрезает часть шоколада на две меньших части, он получает столько же изюминок, сколько было на исходной части.

Бонни хочет заплатить Петру как можно меньше. Она знает, сколько изюминок находится на каждом из $N \cdot M$ кусочков. Она может выбрать порядок, в котором даёт Петру остающиеся части, и она также может говорить Петру, какие именно разрезы делать (горизонтальные или вертикальные) и в каком именно месте. Помогите Бонни решить, как разрезать плитку шоколада на одиночные кусочки таким образом, чтобы расплатиться с Петром как можно меньшим количеством изюминок.

ЗАДАНИЕ

Напишите программу, которая по заданному количеству изюминок на каждом из одиночных кусочков определяет минимальное количество изюминок, которым Бонни должна расплатиться с Петром.

ОГРАНИЧЕНИЯ

$1 \leq N, M \leq 50$ Количество кусочков вдоль каждой стороны плитки шоколада.
 $1 \leq R_{k,p} \leq 1000$ Количество изюминок на кусочке в k -й строке и p -м столбце.

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать из стандартного потока ввода следующие данные:

- Первая строка содержит два целых числа N и M , разделённые одним пробелом.
- Последующие N строк описывают, сколько изюминок находится на каждом кусочке шоколада. k -я из этих N строк описывает k -ю строку плитки. Каждая такая строка содержит M целых чисел, разделённых одиночными пробелами. Эти целые числа описывают кусочки

в соответствующей строке плитки слева направо. p -е из чисел на k -й строке (среди этих N строк) сообщает, сколько изюминок находится на кусочке, размещённом в k -й строке и p -м столбце.

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна писать в стандартный поток вывода одну строку, содержащую одно целое число: минимальное количество изюминок, которым Бонни должна расплатиться с Петром.

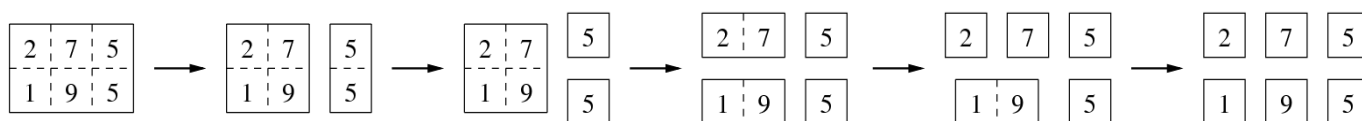
СИСТЕМА ОЦЕНКИ

Для набора тестов общей стоимостью 25 баллов N и M не будут превосходить 7.

ПРИМЕР

Пример ввода	Пример вывода
2 3 2 7 5 1 9 5	77

Один из многих возможных способов достигнуть стоимости, составляющей 77 изюминок такой:



Первый разрез, который Бонни просит сделать Петра, разделяет третий столбец от оставшейся части плитки шоколада. Бонни должна заплатить за это Петру 29 изюминок.

Затем Бонни даёт Петру меньшую из двух частей – ту, которая состоит из двух кусочков по 5 изюминок в каждом, и просит Петра разрезать эту часть на две в обмен на 10 изюминок.

После этого, Бонни даёт Петру большую оставшуюся часть, в которой остались кусочки с 2, 7, 1 и 9 изюминками соответственно. Бонни просит Петра разрезать её по горизонтали, отделяя первую строку от второй, и платит 19 изюминок.

После этого Бонни просит Петра разрезать левую верхнюю часть, платя 9 изюминок. Наконец, Бонни просит Петра разрезать левую нижнюю часть, платя 10 изюминок.

Общая стоимость составляет $29 + 10 + 19 + 9 + 10 = 77$ изюминок. Не существует способа разрезать эту плитку шоколада на составляющие её 6 кусочков за меньшую стоимость.

ПАРКОВКА

Full Feedback Problem

Парковка имеет N мест, пронумерованных от 1 до N включительно. Парковка открывается пустой каждое утро и работает на протяжении дня следующим образом. Когда автомобиль приезжает на парковку, парковщик проверяет, есть ли свободные места. Если таковых нет, автомобиль ожидает возле въезда до тех пор, пока освободится какое-то место. Если есть свободное место, или как только оно освобождается, автомобиль занимает свободное парковочное место. Если есть несколько свободных парковочных мест, автомобиль занимает место с наименьшим номером. Когда приезжают парковаться другие автомобили, но уже есть ожидающий автомобиль, они выстраиваются в очередь на въезде в том порядке, в котором приехали. После того, как освобождается парковочное место, его занимает первый автомобиль из очереди (то есть тот, который прибыл парковаться первым).

Стоимость парковки одного автомобиля в долларах определяется как произведение веса этого автомобиля в килограммах на тариф его парковочного места. Стоимость парковки автомобиля не зависит от того, сколько времени этот автомобиль находится на парковке.

Парковщик знает, что сегодня на парковку приедет M автомобилей, и он знает порядок их приезда и отъезда. Помогите ему подсчитать, сколько долларов он сегодня заработает.

ЗАДАНИЕ

Напишите программу, которая по заданным тарифам парковочных мест, весам автомобилей и порядку, в котором автомобили приезжают и уезжают, определяет доход парковки в долларах.

ОГРАНИЧЕНИЯ

$1 \leq N \leq 100$	Количество парковочных мест
$1 \leq M \leq 2,000$	Количество автомобилей
$1 \leq R_s \leq 100$	Тариф парковочного места с номером s в долларах за килограмм
$1 \leq W_k \leq 10,000$	Вес автомобиля с номером k в килограммах

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать со стандартного потока ввода такие данные:

- Первая строка содержит целые числа N и M , разделенные пробелом.
- Последующие N строк описывают тарифы парковочных мест, s -ая из этих строк содержит одно целое число R_s – тариф парковочного места с номером s в долларах за килограмм.
- Последующие M строк описывают веса автомобилей. Автомобили пронумерованы в произвольном порядке от 1 до M включительно, k -ая из этих M строк содержит одно целое число W_k – вес автомобиля с номером k в килограммах.
- Последующие $2 \cdot M$ строк описывают приезд и выезд всех автомобилей в хронологическом порядке. Положительное целое число i показывает, что автомобиль с номером i приезжает на парковку. Отрицательное целое число $-i$ показывает, что автомобиль с номером i уезжает с парковки. Никакой автомобиль не выезжает с парковки до своего приезда, и все автомобили от 1 до M включительно появятся в этой последовательности строк ровно 2 раза, один раз как

приезжающий, и второй – как выезжающий. К тому же, никакой из автомобилей не выедет с парковки, пока не займет место на парковке (то есть, никакой автомобиль не уедет пока стоит в очереди).

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна записать в стандартный поток вывода одну строку, содержащую одно целое число: общее количество долларов, которое заработает сегодня парковщик.

СИСТЕМА ОЦЕНКИ

Для набора тестов общей стоимостью 40 баллов всегда будет хотя бы одно свободное парковочное место для каждого приезжающего автомобиля. В этих тестах никакой автомобиль не должен будет ожидать освобождения парковочного места.

ПРИМЕРЫ

Пример ввода	Пример вывода	Комментарий
3 4 2 3 5 200 100 300 800 3 2 -3 1 4 -4 -2 -1	5300	Автомобиль с номером 3 занимает место с номером 1 и платит $300 * 2 = 600$ долларов. Автомобиль с номером 2 занимает место с номером 2 и платит $100 * 3 = 300$ долларов. Автомобиль с номером 1 занимает место с номером 1 (которое освободил автомобиль с номером 3) и платит $200 * 2 = 400$ долларов. Автомобиль с номером 4 занимает место с номером 3 (последнее оставшееся) и платит $800 * 5 = 4000$ долларов.

Пример ввода	Пример вывода	Комментарий
2 4 5 2 100 500 1000 2000 3 1 2 4 -1 -3 -2 -4	16200	Автомобиль с номером 3 занимает место с номером 1 и платит $1000 * 5 = 5000$ долларов. Автомобиль с номером 1 занимает место с номером 2 и платит $100 * 2 = 200$ долларов. Автомобиль с номером 2 приезжает и должен ждать на въезде. Автомобиль с номером 4 приезжает и должен ждать на въезде за автомобилем с номером 2. Когда автомобиль с номером 1 освобождает свое парковочное место, автомобиль с номером 2 занимает это место и платит $500 * 2 = 1000$ долларов. Когда автомобиль с номером 3 освобождает свое парковочное место, автомобиль с номером 4 занимает это место и платит $2000 * 5 = 10000$ долларов.

МЕДВЕДЬ МИША

Медведь Миша нашел маленькое сокровище – спрятанный горшочек меда! Он с удовольствием поедает мед, но вдруг одна пчела его заметила и забила тревогу. Миша знает, что после этого пчелы вылетят из своих ульев и будут разлетаться вокруг, чтобы настичь его. Он знает, что нужно бросать горшочек и быстро идти домой, но мед так сладок, что Миша не хочет бросать его есть слишком рано. Помогите Мише определить последний возможный момент, когда он может прекратить есть мед.

Лес представлен картой в виде квадратной сетки, которая состоит из $N \times N$ единичных ячеек, стороны которых параллельны направлениям «север-юг» и «запад-восток». Каждая ячейка леса занята либо деревом, либо травой, либо ульем, либо Мишиным домом. Две ячейки называются смежными, если одна из них находится непосредственно к северу, югу, востоку или западу от другой, но не по диагонали. Миша неповоротлив, поэтому он может перемещаться только в смежную ячейку. Миша может перемещаться только по ячейкам с травой и не может перемещаться по ячейкам с деревьями или ульями. Также он не может перемещаться больше, чем на S ячеек в минуту.

В момент, когда прозвучала тревога, Миша находится в ячейке с травой, где он нашел горшочек с медом, а все пчелы – в ячейках, где расположены ульи (в лесу может быть больше одного улья). С этого момента, на протяжении каждой следующей минуты происходят следующие события в таком порядке:

- Если Миша все еще ест мед, он решает, будет ли он продолжать есть или будет уходить. Если он продолжает есть мед – он не перемещается всю минуту. Иначе, он немедленно уходит и перемещается по лесу не более чем на S ячеек, как описано выше. Миша не может брать с собой мед, и как только он ушел, он уже не может его есть.
- Как только Миша заканчивает есть или перемещаться в течение минуты, пчелы разлетаются на одну ячейку дальше, занимая только ячейки с травой. Точнее, пчелы разлетаются во все ячейки с травой, смежные с любой ячейкой, где уже есть пчелы. Как только в ячейке появляются пчелы, они там остаются навсегда (пчелы не перемещаются, а распространяются).

Другими словами, пчелы разлетаются так: когда звучит тревога, пчелы находятся в ячейках, где расположены ульи. В конце первой минуты они занимают все ячейки с травой, смежные с ульями, и остаются в тех ячейках, где расположены ульи. В конце второй минуты пчелы дополнительно занимают все ячейки с травой, смежные со смежными с ульями ячейками, и так далее. Имея достаточно времени, пчелы займут все ячейки с травой, которые они могут достичь.

Ни Миша, ни пчелы не могут покидать пределы леса. Также обратите внимание, что согласно описанным правилам, Миша ест мед целое число минут.

Пчелы настигают Мишу, если в какой-то момент времени Миша оказывается в ячейке, занятой пчелами.

ЗАДАНИЕ

Напишите программу, которая по карте леса определяет наибольшее количество минут, на протяжении которых Миша может продолжать есть мед в своем исходном расположении, все еще имея возможность попасть домой до того, как пчелы его настигнут.

ОГРАНИЧЕНИЯ

- | | |
|------------------------|--|
| $1 \leq N \leq 800$ | Размер (длина стороны) карты леса |
| $1 \leq S \leq 1\,000$ | Максимальное количество перемещений, которые может делать Миша в каждую минуту |

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать со стандартного потока ввода такие данные:

- Первая строка содержит целые числа **N** и **S**, разделенные пробелом.
- Последующие **N** строк задают карту леса. Каждая из этих строк содержит **N** символов, каждый символ задает одну ячейку на сетке. Возможные символы и их значения описаны ниже:

T обозначает ячейку с деревом

G обозначает ячейку с травой

M обозначает начальное расположение Миши и горшочка меда в ячейке с травой

D обозначает ячейку где расположен Мишин дом, в который Миша может попасть, а пчелы не могут

H обозначает ячейку с ульем

ПРИМЕЧАНИЕ: Гарантируется, что карта леса содержит ровно одну букву **M**, ровно одну букву **D** и, по крайней мере, одну букву **H**. Также гарантируется, что существует последовательность смежных ячеек **G**, которые соединяют ячейку с начальным расположением Миши и ячейку, где расположен Мишин дом, так же как и последовательность смежных ячеек **G**, которые соединяют хотя бы одну из ячеек с ульем с ячейкой с горшочком (то есть, с ячейкой с Мишиным начальным расположением). Последовательности могут быть и с длиной, равной нулю, в случае, если ячейка с Мишиным домом или ячейка с ульем являются смежными с ячейкой с начальным расположением Миши. Также заметьте, что пчелы не могут распространяться через ячейку с Мишиным домом. Для пчел она – как ячейка с деревом.

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна записать в стандартный поток вывода единственную строку, содержащую одно целое число: максимально возможное количество минут, на протяжении которых Миша может продолжать есть мед в начальном расположении, имея возможность безопасно вернуться домой.

Если Миша не имеет возможность вернуться домой до того, как пчелы его настигнут, ваша программа должна вывести отрицательное число **-1**.

СИСТЕМА ОЦЕНКИ

Для набора тестов общей стоимостью 40 баллов, **N** не будет превышать 60.

ПРИМЕРЫ

Пример ввода	Пример вывода
<pre> 7 3 TTTTTTT TGGGGGT TGGGGGT MGGGGGD TGGGGGT TGGGGGT TNNNNHT </pre>	<pre> 1 </pre>

Миша может есть мед одну минуту, после чего может идти последующие 2 минуты направо (→→→) к своему дому по кратчайшему пути, безопасному от пчел.

Пример ввода	Пример вывода
<pre> 7 3 TTTTTTT TGGGGGT TGGGGGT MGGGGGD TGGGGGT TGGGGGT TGNHGGT </pre>	<pre> 2 </pre>

Миша может есть мед две минуты, после чего может сделать шаги →↑→ на протяжении третьей минуты, после чего шаги →→→ на протяжении четвертой минуты и шаги ↓→ на протяжении пятой минуты.

РЕГИОНЫ

Агентство по Развитию Регионов Организации Объединённых Наций (АРРООН) имеет хорошую организационную структуру. В АРРООН работают N человек, каждый из которых приехал из одного из R различных географических регионов мира. Регионы пронумерованы от 1 до R включительно в произвольном порядке. Работники имеют номера от 1 до N включительно в порядке убывания их возрастов. При этом работник с номером 1 (председатель) является самым старшим из них. Каждый работник, кроме председателя, имеет одного непосредственного начальника. Начальник всегда старше, чем работник, которым он руководит.

Будем говорить, что работник A является менеджером работника B , если работник A является либо непосредственным начальником работника B , либо менеджером непосредственного начальника работника B . Например, председатель является менеджером любого другого работника. Кроме того, очевидно, что никакие два работника не могут быть менеджерами друг друга.

К сожалению, Бюро Исследований Организации Объединённых Наций (БИООН) недавно получило некоторое количество жалоб, что АРРООН имеет несбалансированную структуру, более выгодную для одних регионов и менее выгодную для других. Чтобы расследовать эти случаи, БИООН собирается создать компьютерную систему, которой будет задаваться описание структуры АРРООН и которая будет отвечать на запросы в следующей форме: по двум регионам r_1 и r_2 найти количество таких пар работников (e_1, e_2) , что работник e_1 приехал из региона r_1 , работник e_2 приехал из региона r_2 , и e_1 является менеджером e_2 . У каждого запроса два параметра: номера регионов r_1 и r_2 . Результат выполнения запроса – одно целое число: количество различных пар (e_1, e_2) , которые удовлетворяют вышеописанным условиям.

ЗАДАНИЕ

Напишите программу, которая по заданным регионам, из которых приехали работники агентства, а также информации о том, кто является чьим непосредственным начальником, интерактивно отвечает на вопросы в вышеописанной форме.

ОГРАНИЧЕНИЯ

$1 \leq N \leq 200\,000$	Количество работников
$1 \leq R \leq 25\,000$	Количество регионов
$1 \leq Q \leq 200\,000$	Количество запросов, на которые должна ответить ваша программа
$1 \leq H_k \leq R$	Регион, из которого приехал работник с номером k (для $1 \leq k \leq N$)
$1 \leq S_k < k$	Непосредственный начальник работника с номером k (для $2 \leq k \leq N$)
$1 \leq r_1, r_2 \leq R$	Регионы, по которым делается запрос

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать из стандартного потока ввода следующие данные:

- Первая строка содержит в указанном порядке целые числа N , R и Q , разделённые одиночными пробелами.
- Последующие N строк описывают N работников агентства в порядке от старших к младшим. k -я из этих N строк описывает работника с номером k . Первая из этих строк (то есть, та, которая описывает председателя) содержит одно целое число H_1 – номер региона, из которого приехал председатель. Каждая из последующих $(N-1)$ строк содержит два целых числа, разделённых одним пробелом: целое число S_k – номер непосредственного начальника для работника с номером k и целое число H_k – номер региона, из которого приехал работник с номером k .

ИНТЕРАКТИВНОСТЬ

После прочтения входных данных ваша программа должна начать в интерактивном режиме читать запросы из стандартного потока ввода и выводить результаты обработки запросов в стандартный поток вывода. На каждый из Q запросов необходимо отвечать отдельно, то есть, ваша программа должна вывести результат обработки принятого запроса до получения следующего запроса.

Каждый запрос представлен в одной строке стандартного потока ввода и состоит из двух целых чисел – номеров регионов r_1 и r_2 . Эти числа разделены одним пробелом.

Ответом на каждый запрос является одна строка, выведенная в стандартный поток вывода. Эта строка должна содержать одно целое число – количество пар работников АРРООН (e_1, e_2) таких, что работник с номером e_1 приехал из региона с номером r_1 , работник с номером e_2 приехал из региона с номером r_2 и работник с номером e_1 является менеджером работника с номером e_2 .

ЗАМЕЧАНИЕ: Тесты будут такими, что правильный ответ на любой запрос, задаваемый в стандартном потоке ввода, всегда будет меньше 1 000 000 000.

ЕЩЁ ОДНО ВАЖНОЕ ЗАМЕЧАНИЕ: Чтобы правильно взаимодействовать с системой оценки, вам необходимо сбрасывать буфер стандартного потока ввода после ответа на каждый запрос. Кроме того, необходимо избегать операций со входным потоком, которые могут его заблокировать, таких как, например, `scanf ("%d\n")`. Пожалуйста, посмотрите страницы с технической информацией, на которых содержится описание того, как правильно это делать.

ОЦЕНИВАНИЕ

1. Для набора тестов общей стоимостью 30 баллов R не будет превосходить 500.
2. Для набора тестов общей стоимостью 55 баллов ни из какого региона не приехало более 500 работников.
3. Тесты, в которых выполняются оба условия 1 и 2, имеют стоимость 15 баллов.
4. Тесты, в которых выполняется хотя бы одно из условий 1 и 2, имеют стоимость 70 баллов.

ПРИМЕР

Пример ввода	Пример вывода
6 3 4	
1	
1 2	
1 3	
2 3	
2 3	
5 1	
1 2	
1 3	1 [сброс буфера вывода]
2 3	3 [сброс буфера вывода]
3 1	2 [сброс буфера вывода]
	1 [сброс буфера вывода]

ТЕСТИРОВАНИЕ

При тестировании вашего решения с использованием интерфейса “test” проверяющей системы входной файл, который вы передаёте, должен содержать как входные данные, так и запросы, как показано в примере ввода выше.

КОММИВОЯЖЁР

Коммивояжёр решил, что определение оптимального расписания его передвижения по дорогам – это неразрешимая вычислительная задача. Поэтому он занялся бизнесом, используя только передвижение по реке Данубе, имеющей прямое русло. У него есть моторная лодка, которая может довезти его с любого места на реке до любого другого места реки, не тратя на это абсолютно никакого времени. К сожалению, лодка потребляет очень много горючего. На каждый метр проезда против течения реки (к истоку) потребляется горючее стоимостью U долларов, а на каждый метр проезда по течению реки (от истока) – D долларов.

Коммивояжёр хочет посетить N ярмарок, которые будут проводиться в различных местах, расположенных вдоль реки. Каждая ярмарка длится ровно один день. Для каждой ярмарки с номером X коммивояжёр знает её дату проведения T_x , измеряемую в днях с момента покупки коммивояжёром лодки. Ему также известны место расположения ярмарки L_x , измеряемое в метрах от истока реки в направлении вниз по течению, и количество долларов M_x , которые он выручит, если посетит эту ярмарку. Он начинает и заканчивает своё путешествие в своем доме, расположенном в S метрах от истока реки в направлении вниз по течению.

Помогите коммивояжёру выбрать, какие ярмарки ему следует посетить (если требуется) и в каком порядке, чтобы получить максимальную общую прибыль по окончании путешествия. Общая прибыль коммивояжёра вычисляется как разность между суммой долларов, которую он выручил от посещения ярмарок, и суммой долларов, потраченной им на горючее при передвижении вверх и вниз по реке.

Имейте в виду, что если ярмарка A проводится раньше ярмарки B , то коммивояжёр может посетить их только в порядке их дат проведения (то есть, он не может посетить ярмарку B раньше, чем ярмарку A). Если две ярмарки проводятся в один день, то коммивояжёр может посетить эти две ярмарки в любом порядке в тот же день. Нет ограничения на количество ярмарок, посещаемых в один день, но, естественно, коммивояжёр не может посетить одну ярмарку дважды, получив двойную прибыль. При этом он может проезжать через места уже посещённых ярмарок, не получая никакой прибыли.

ЗАДАНИЕ

Напишите программу, которая по заданным датам проведения ярмарок, их местам расположения и по выручке, получаемой от посещения ярмарок, а также месту расположения дома коммивояжёра и стоимостям переезда, определит максимальную возможную прибыль, которую коммивояжёр может получить по окончании своего путешествия.

ОГРАНИЧЕНИЯ

$$1 \leq N \leq 500\,000$$

$$1 \leq D \leq U \leq 10$$

$$1 \leq S \leq 500\,001$$

$$1 \leq T_k \leq 500\,000$$

$$1 \leq L_k \leq 500\,001$$

$$1 \leq M_k \leq 4\,000$$

Количество ярмарок

Стоимость перемещения на один метр вверх по течению реки (U) и вниз по течению реки (D)

Место расположения дома коммивояжёра

День, в который проводится ярмарка с номером k

Место расположения ярмарки с номером k

Количество долларов, которые коммивояжёр получит в результате посещения ярмарки с номером k

ВХОДНЫЕ ДАННЫЕ

Ваша программа должна читать из стандартного потока ввода следующие данные:

- Первая строка содержит в указанном порядке целые числа N , U , D и S , разделённые одиночными пробелами.
- Последующие N строк описывают N ярмарок в произвольном порядке, k -я из этих N строк описывает k -ю ярмарку и содержит три целых числа, разделённых одиночными пробелами: день проведения ярмарки T_k , её место расположения L_k , и выручку, которую может получить коммивояжёр от посещения этой ярмарки M_k .

ПРИМЕЧАНИЕ: Все места расположения, указанные во входных данных, будут различны. В частности, никакие две ярмарки не будут проводиться в одном месте, и никакая ярмарка не будет проводиться в том же месте, где находится дом коммивояжёра.

ВЫХОДНЫЕ ДАННЫЕ

Ваша программа должна вывести в стандартный поток вывода одну строку, содержащую одно целое число: максимальную прибыль, которую коммивояжёр может получить по окончании своего путешествия.

СИСТЕМА ОЦЕНКИ

1. Для набора тестов общей стоимостью 60 баллов никакие две ярмарки не будут проводиться в один и тот же день.
2. Для набора тестов общей стоимостью 40 баллов ни одно из чисел во входных данных не будет превосходить 5 000.
3. Тесты, в которых выполняются оба условия 1 и 2, имеют стоимость 15 баллов.
4. Тесты, в которых выполняется хотя бы одно из условий 1 и 2, имеют стоимость 85 баллов.

ПРИМЕР

Пример ввода	Пример вывода
4 5 3 100 2 80 100 20 125 130 10 75 150 5 120 110	50

Оптимальное расписание заключается в посещении ярмарок с номерами 1 и 3 (места расположения которых – 80 и 75 метров от истока реки вниз по течению). Последовательность событий и прибыль, получаемая в результате этих событий, таковы:

- Коммивояжёр проплывает 20 метров вверх по течению и тратит на это 100 долларов. Прибыль к этому моменту является отрицательной и составляет **-100** долларов.
- Он посещает ярмарку с номером 1 и выручает 100 долларов. Прибыль к этому моменту составляет **0** долларов.
- Он проплывает 5 метров вверх за 25 долларов. Прибыль к этому моменту является отрицательной и составляет **-25** долларов.
- Он посещает ярмарку с номером 3, где он выручает 150 долларов. Прибыль к этому моменту составляет **125** долларов.
- Он проплывает 25 метров вниз, чтобы вернуться домой, на что тратит 75 долларов. Прибыль по окончании путешествия составляет **50** долларов.