

Задача BFS. Лабиринт

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

В одном из уровней компьютерной игры вы попали в лабиринт, состоящий из n строк, каждая из которых содержит m клеток. Каждая клетка либо свободна, либо занята препятствием. Стартовая клетка находится в строке r и столбце c . За один шаг вы можете переместиться на одну клетку вверх, влево, вниз или вправо, если она не занята препятствием. Вы не можете перемещаться за границы лабиринта.

К сожалению, ваша клавиатура крайне близка к поломке, поэтому вы можете переместиться влево не более x раз и вправо не более y раз. При этом ограничений на перемещения вверх и вниз нет, поскольку клавиши, используемые для движения вверх и вниз, всё ещё в идеальном состоянии.

Теперь вы для каждой клетки поля решили установить, можно ли выбрать такую последовательность нажатий, которая приведёт вас из стартовой в эту клетку. Посчитайте, сколько клеток поля обладают таким свойством.

Формат входных данных

Первая строка содержит два целых числа n, m ($1 \leq n, m \leq 2000$) — количество строк и столбцов в лабиринте, соответственно.

Вторая строка содержит два целых числа r, c ($1 \leq r \leq n, 1 \leq c \leq m$) — номер строки и столбца, на пересечении которых расположена стартовая клетка.

Третья строка содержит два целых числа x, y ($0 \leq x, y \leq 10^9$) — максимальное количество перемещений влево и вправо, соответственно.

Следующие n строк содержат описание лабиринта. Каждая из этих строк имеет длину m и состоит только из символов '.' и '*'. В i -й строке j -й символ соответствует клетке лабиринта с номерами строки и столбца i и j , соответственно. Символ '.' соответствует свободной клетке лабиринта, а символ '*' — клетке с препятствием.

Гарантируется, что стартовая клетка не занята препятствием.

Формат выходных данных

Выведите одно число — количество клеток лабиринта, достижимых из стартовой, включая её саму.

Примеры

стандартный ввод	стандартный вывод
4 5 3 2 1 2***. ...** *.....	10
5 5 5 4 3 1 **... **.*. ...*. .***.	16

Задача VIN. Ярость Битвы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	64 мегабайта

Герой по имени Магина сражается с группой из n монстров с помощью легендарного топора, известного как Ярость Битвы. Каждый из монстров имеет a_i очков здоровья. Каждым ударом топора Магина уменьшает здоровье того, кого он ударил, на p очков, при этом уменьшая здоровье всех остальных монстров на q очков. Монстр умирает, когда у него остается 0 или менее очков здоровья. Магина хочет при каждом ударе выбирать цель таким образом, чтобы убить всех монстров за минимальное количество ударов. Требуется определить это количество.

Формат входных данных

В первой строке содержатся три целых числа через пробел: n , p и q ($1 \leq n \leq 10000$, $1 \leq q \leq p \leq 10^9$) — количество монстров, урон по цели и урон по всем остальным соответственно.

Во второй строке содержатся n целых чисел через пробел: a_i ($1 \leq a_i \leq 10^9$) — количество очков здоровья у каждого из монстров.

Формат выходных данных

Выведите единственное целое число — минимальное количество ударов, за которое Магина сможет убить всех монстров.

Примеры

стандартный ввод	стандартный вывод
2 3 2 5 5	2
3 5 3 17 13 14	5

Задача СРР. Демид огонь!!11!!1!!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

2031 год. Прогресс ушёл далеко вперёд, и теперь школьникам не нужно собираться на линейку — во время музыкального подъёма расписание и объявления транслируются непосредственно им в мозг. К сожалению, до столовой «Берендеевых полян» прогресс доходит не так быстро, и у дверей всё ещё скапливается очередь. Демид придумал решение проблемы: школьники выстраиваются перед столовой в шеренги и по очереди целыми шеренгами запускаются внутрь. К сожалению, школьники всё ещё такие же неорганизованные и подходят к столовой не все вместе, а небольшими группами и встают всей группой в одну из шеренг. Демиду это не нравится, но поделать нечего, так что он придумал систему для запуска лкшат в столовую.

Школьники одной параллели в столовой сидят рядом, и Демид находит логичным запускать большое число лкшат из одной параллели пораньше. Поэтому он присваивает каждой шеренге приоритет, равный сумме квадратов длин подряд идущих отрезков школьников из одной параллели, и запускает в столовую шеренгу с максимальным приоритетом, а при равенстве — с минимальным номером. Шеренги пронумерованы числами от 1 до n . Изначально каждая из шеренг пуста.

Поддерживать такую систему приоритетов сложно, ведь школьники постоянно приходят, и приоритеты изменяются. Так что Демид просит вас помочь с этой трудной задачей.

Формат входных данных

В первой строке даны два числа n и q — число шеренг и число событий ($1 \leq n, q \leq 100\,000$).

Каждая из следующих q строк описывает очередное событие.

Первое число в строке равно 1 или 2 и означает тип события.

Событие типа 1 означает появление группы школьников. За единицей в этой же строке следуют два целых числа x и k ($1 \leq x \leq n$, $1 \leq k \leq 100\,000$) — номер шеренги, в которую встали только что подошедшие школьники, и их количество. После них в той же строке находятся k целых чисел p_1, \dots, p_k ($1 \leq p_i \leq 100\,000$) — номера параллелей, в которых учатся эти школьники. Они добавляются в конец шеренги именно в таком порядке.

Событие типа 2 означает запуск наиболее приоритетной шеренги в столовую. Такая строка не содержит иной информации.

Сумма величин k по всем запросам типа 1 не превосходит 10^6 .

Формат выходных данных

Для каждого события типа 2 в отдельной строке необходимо вывести одно число — номер шеренги, которую Демид пустит в столовую.

Примеры

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

Замечание

У всех трёх шеренг одинаковый приоритет, так как у первой и третьей шеренги приоритет равен 2^2 , а у второй — $1^2 + 1^2 + 1^2 + 1^2$.

Задача DFS. Многоэтажки и пожилая черепашка

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Пожилая черепашка Полина очень любит прямоугольное клетчатое болото Осуждаево, в левом верхнем углу которого обитает уже много-много лет. Но вот незадача: многоэтажная застройка добралась и до её родного дома. Поэтому пожилая Полина решила подать жалобу в местные управляющие болотные органы. Но, к сожалению, здание правительства находится достаточно далеко от её привычного места обитания: в правом нижнем углу.

Долгие путешествия не пугают Полину, но буквально вчера она прочитала объявление, в котором подробно описывался план застройки Осуждаево: там были перечислены клетки болота в порядке их захвата многоэтажками; план составлен так, что каждая многоэтажка занимает ровно одну клетку. Полина — очень целеустремлённая пожилая черепашка, поэтому по пути к зданию правительства она умеет двигаться только либо вниз, либо вправо, совершая шаги, которые только приближают её к своей цели, но она не может находиться в клетке, которая захвачена многоэтажкой.

Помогите Полине узнать, какое максимальное количество многоэтажек может быть построено до того, как она потеряет возможность добраться до здания правительства и подать свою жалобу и осудить застройку Осуждаева.

Формат входных данных

В первой строке даны три целых числа n , m и k ($2 \leq n, m \leq 300$; $1 \leq k \leq n \cdot m - 1$) — размеры болота и количество клеток, которые будут застроены многоэтажками.

В следующих k строках даны по два целых числа x и y ($1 \leq x \leq n$; $1 \leq y \leq m$) — координаты клеток, которые будут последовательно застроены (в том порядке, в котором даны). Гарантируется, что все клетки различны. Также гарантируется, что стартовая клетка так и не будет застроена.

Формат выходных данных

В единственной строке выведите номер многоэтажки, после постройки которой пожилая Полина перестанет иметь возможность подать жалобу в правительство Осуждаево, или -1 , если этого не случится.

Примеры

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

Задача DPR. Число возрастающих подпоследовательностей

Имя входного файла: `subseq.in`
Имя выходного файла: `subseq.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Задана последовательность из n чисел a_1, a_2, \dots, a_n . Подпоследовательностью длины k этой последовательности называется набор индексов i_1, i_2, \dots, i_k , удовлетворяющий неравенствам $1 \leq i_1 < i_2 < \dots < i_k \leq n$. Подпоследовательность называется возрастающей, если выполняются неравенства $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности a_1, \dots, a_n . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на $10^9 + 7$.

Формат входных данных

Первая строка входного файла содержит целое число n ($1 \leq n \leq 1000$). Вторая строка входного файла содержит n целых чисел: a_1, a_2, \dots, a_n . Все a_i не превосходят 10^9 по абсолютной величине.

Формат выходных данных

В выходной файл выведите ответ на задачу.

Примеры

subseq.in	subseq.out
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8

Задача DSU. Ася и котята

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Ася очень любит животных. Недавно она приобрела n котят, пронумеровала их от 1 до n и поселила в вольере. Вольер представляет собой ряд из n ячеек, пронумерованных от 1 до n слева направо. Соседние ячейки разделены сетчатыми перегородками, всего в вольере $n - 1$ перегородок. Изначально в каждой ячейке поселился ровно один котёнок с некоторым номером.

Наблюдая за котятами, Ася заметила, что они очень дружелюбны и некоторые пары живущих в соседних ячейках котят очень хотят играть друг с другом. Чтобы не лишать их этого удовольствия, Ася стала вынимать перегородки между соседними ячейками. В частности, в i -й день, Ася:

- Обратила внимание, что котята x_i и y_i , живущие в соседних ячейках, хотят играть вместе.
- Удаляла перегородку между этими ячейками, превращая их в одну, в которой оказывались все котята из двух прежних ячеек.

Поскольку Ася ни разу возвращала перегородки, через $n - 1$ день вольер стал единой ячейкой, в которой оказались все котята.

Для каждого дня, Ася помнит номера котят x_i и y_i , которые хотели играть вместе, однако она не помнит как котята были поселены в ячейки изначально. Помогите ей найти какое-нибудь возможное начальное расселение котят по n ячейкам.

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 150\,000$) — количество котят.

Каждая из следующих $n - 1$ строк содержит пары целых чисел x_i, y_i ($1 \leq x_i, y_i \leq n, x_i \neq y_i$) — номера котят, между ячейками которых была удалена перегородка в очередной день.

Гарантируется, что котята x_i и y_i ещё не находились в одной ячейке по итогам предыдущих объединений.

Формат выходных данных

Для каждой ячейки от 1 до n выведите целое число — номер котёнка от 1 до n , который в ней находился изначально.

Все выведенные числа должны быть различны.

Гарантируется, что существует хотя бы один возможный ответ. В случае, если существует несколько решений, выведите любое из них.

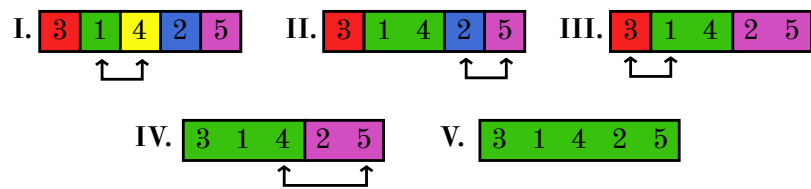
Примеры

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

Замечание

В ответе к примеру приведено одно из нескольких возможных изначально расселений котят.

На картинке ниже показано, как происходило объединение ячеек при таком изначально расселении. Обратите внимание, что котята, желавшие играть вместе в каждый из дней, действительно находились в **соседних ячейках**.



Задача КНР. Олежка и пульт

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Мало кто знает, что во время генерального прогона детского спектакля сломался пульт. До спектакля оставалось крайне мало времени, и привезти новый пульт из Москвы просто не успевали. Казалось бы, нет никакой надежды его починить. Все беспокоились, что спектакль вот-вот будет сорван, но в самый ответственный момент наши админы Булат, Егор и Олежка вызвались собрать новый пульт! Для этого им было необходимо по одной детали каждого из n типов. Каждая деталь производится k_i производителями, и у каждой детали каждого производителя есть собственная цена и качество.

Админы поехали в Судиславль, чтобы купить нужные детали, но у них было с собой только b рублей. Они хотели, чтобы новый пульт прослужил как можно дольше, поэтому старались купить такой набор деталей, чтобы качество пульта было как можно выше (качество пульта определяется качеством его худшей детали).

Когда же детали были куплены, админы с энтузиазмом бросились собирать пульт, забыв про все остальное. Теперь пульт собран, но админы уже не помнят, какие детали они купили, и не понимают, насколько качественный получился пульт. Ваша задача — помочь Олежке понять, какого качества получился пульт, или вывести -1 , если они не могли собрать пульт из деталей, купленных на не более чем b рублей.

Формат входных данных

Первая строка содержит два целых числа n и b ($1 \leq n \leq 10^4, 0 \leq b \leq 10^9$) — количество типов деталей и бюджет админов.

Далее идет n блоков описаний деталей одного типа. Каждый блок начинается со строчки с одним числом — k_i . Следующие k_i строк содержат по два целых числа c_j и p_j ($0 \leq c_j \leq 10^4, 0 \leq p_j \leq 10^9$) — стоимость детали и ее качество.

Сумма всех k_i не превосходит 10^4 .

Формат выходных данных

Вывести одно число — ответ на задачу.

Примеры

stdin	stdout
2 5 2 1 7 3 5 3 1 1 0 8 4 6	7
2 5 1 1 3 2 5 5 10 10	-1

Задача LIN. Все минимумы

Имя входного файла: `minsum.in`
Имя выходного файла: `minsum.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дан массив целых чисел a_1, a_2, \dots, a_n .

Для каждого его подотрезка $[a_L..a_R]$ определим $F(L, R) := \min\{a_L, \dots, a_R\}$.

Найдите

$$\sum_{1 \leq L \leq R \leq n} F(L, R)$$

то есть сумму минимумов всех подотрезков.

Формат входных данных

Первая строка входных данных содержит натуральное число n ($1 \leq n \leq 100\,000$) — размер массива. Во второй строке через пробел заданы элементы массива, все числа целые от -10^6 до 10^6 .

Формат выходных данных

Выведите единственное число — сумму минимумов всех подотрезков массива a .

Примеры

<code>minsum.in</code>	<code>minsum.out</code>
1 5	5
2 -10 1	-19
4 1 2 3 4	20
5 -3 2 -4 1 -5	-52

Задача LOG. Поле боя

Имя входного файла:	battlefield.in
Имя выходного файла:	battlefield.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сегодня в длительной и непримиримой войне между Аарляндией и Берляндией происходит ключевая битва. Командование армии Берляндии собралось в штабе перед специальным радаром, чтобы следить за сражением. Радар показывает неуничтоженные боевые единицы (и их координаты), которые в данный момент находятся на поле боя.

К сожалению, радар этот не очень хорош. Боевые единицы на нем могут исчезать и появляться вновь на том же месте. К тому же, радар не показывает принадлежность боевых единиц армиям (то есть непонятно, свой ли это солдат или чужой). Командование знает порядок событий вида:

- боевая единица возникла на радаре в точке (x, y) ;
- боевая единица исчезла с радара из точки (x, y) .

В общей неразберихе боя командование армии Берляндии с помощью радара иногда пытается понять, где в данный момент находится линия фронта (это нужно для планирования дальнейших боевых действий). Исходя из соображения о том, что силы примерно равны, командование полагает, что линия фронта — это вертикальная прямая вида $x = a + \frac{1}{2}$ (a — неотрицательное целое число), такая, что количество боевых единиц слева от этой прямой **строго равно** количеству боевых единиц справа от этой прямой, и **a максимально** (то есть, если возможных линий фронта существует несколько, руководство хочет знать самую правую из них).

Помогите командованию армии не проиграть эту битву. Напишите программу, которая будет сообщать линию фронта для текущей ситуации на поле боя, когда этого требует руководство армии Берляндии.

Формат входных данных

В первой строке вводится одно целое число n — количество событий ($1 \leq n \leq 10^5$).

В следующих n строках вводятся описания событий трех видов.

- «A x y » — боевая единица возникла на радаре в точке с целочисленными координатами x, y ($1 \leq x, y \leq 10^9$).
- «D x y » — боевая единица исчезла с радара из точки с целочисленными координатами x, y ($1 \leq x, y \leq 10^9$). Гарантируется, что до этого запроса в этой точке на радаре была боевая единица.
- «Q» — командование Берляндии захотело узнать текущую линию фронта.

Формат выходных данных

Для каждого события вида «Q» выведите в отдельной строке **максимальное** неотрицательное целое a такое, что прямая $x = a + \frac{1}{2}$ — это линия фронта. В случае, если линии фронта не существует, или существует бесконечно много вариантов провести линию фронта, выведите «-1».

Примеры

battlefield.in	battlefield.out
12	1
A 1 1	-1
A 2 1	2
Q	2
A 2 2	
A 3 1	
Q	
A 4 1	
A 5 1	
Q	
D 1 1	
D 5 1	
Q	

Задача NTN. Китайская теорема

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Решите в целых числах систему уравнений

$$\begin{cases} x \equiv a \pmod{n} \\ x \equiv b \pmod{m}, \end{cases}$$

где n и m взаимно просты. Среди решений следует выбрать наименьшее неотрицательное число.

Формат входных данных

Входной файл содержит четыре целых числа a , b , n и m ($1 \leq n, m \leq 10^6$, $0 \leq a < n$, $0 \leq b < m$).

Формат выходных данных

В выходной файл выведите искомое наименьшее неотрицательное число x .

Примеры

stdin	stdout
1 0 2 3	3
3 2 5 9	38

Задача SEG. Мега-инверсии

Имя входного файла: `mega.in`
Имя выходного файла: `mega.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Инверсией в перестановке p_1, p_2, \dots, p_N называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовём мега-инверсией в перестановке p_1, p_2, \dots, p_N тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Напишите алгоритм для быстрого подсчёта количества мега-инверсий в перестановке.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 100\,000$). Следующие N чисел описывают перестановку: p_1, p_2, \dots, p_N ($1 \leq p_i \leq N$), все p_i попарно различны. Числа разделяются переводами строк.

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке p_1, p_2, \dots, p_N .

Примеры

mega.in	mega.out
4	4
4	
3	
2	
1	

Задача STR. Подстроки

Имя входного файла: `substr.in`
Имя выходного файла: `substr.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано K строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leq K \leq 10$). В следующих K строках — собственно K строк (длины строк от 1 до 10 000).

Формат выходных данных

Наибольшая общая подстрока. Гарантируется, что она не пуста.

Примеры

<code>substr.in</code>	<code>substr.out</code>
3 abacaba mycabarchive acabistrue	cab