

Простые задачи

Задача А. Сумма

Имя входного файла: `sum.in`
Имя выходного файла: `sum.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан массив из N элементов, нужно научиться находить сумму чисел на отрезке.

Формат входного файла

Первая строка входного файла содержит два целых числа N и K — число чисел в массиве и количество запросов. ($1 \leq N \leq 100\,000$), ($0 \leq K \leq 100\,000$). Следующие K строк содержат запросы

- `A i x` — присвоить i -му элементу массива значение x ($1 \leq i \leq n$, $0 \leq x \leq 10^9$)
- `Q l r` — найти сумму чисел в массиве на позициях от l до r . ($1 \leq l \leq r \leq n$)

Изначально в массиве живут нули.

Формат выходного файла

На каждый запрос вида `Q l r` нужно вывести единственное число — сумму на отрезке.

Примеры

<code>sum.in</code>	<code>sum.out</code>
<code>5 9</code>	<code>0</code>
<code>A 2 2</code>	<code>2</code>
<code>A 3 1</code>	<code>1</code>
<code>A 4 2</code>	<code>2</code>
<code>Q 1 1</code>	<code>0</code>
<code>Q 2 2</code>	<code>5</code>
<code>Q 3 3</code>	
<code>Q 4 4</code>	
<code>Q 5 5</code>	
<code>Q 1 5</code>	

Задача В. И снова сумма...

Имя входного файла: `sum2.in`
Имя выходного файла: `sum2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Реализуйте структуру данных, которая поддерживает множество S целых чисел, с которым разрешается производить следующие операции:

- `add(i)` — добавить в множество S число i (если он там уже есть, то множество не меняется);
- `sum(l, r)` — вывести сумму всех элементов x из S , которые удовлетворяют неравенству $l \leq x \leq r$.

Формат входного файла

Исходно множество S пусто. Первая строка входного файла содержит n — количество операций ($1 \leq n \leq 300\,000$). Следующие n строк содержат операции. Каждая операция имеет вид либо `+ i`, либо `? l r`. Операция `? l r` задает запрос `sum(l, r)`.

Если операция `+ i` идет во входном файле в начале или после другой операции `+`, то она задает операцию `add(i)`. Если же она идет после запроса `?`, и результат этого запроса был y , то выполняется операция `add((i + y) mod 109)`.

Во всех запросах и операциях добавления параметры лежат в интервале от 0 до 10^9 .

Формат выходного файла

Для каждого запроса выведите одно число — ответ на запрос.

Пример

<code>sum2.in</code>	<code>sum2.out</code>
<code>6</code>	<code>3</code>
<code>+ 1</code>	<code>7</code>
<code>+ 3</code>	
<code>+ 3</code>	
<code>? 2 4</code>	
<code>+ 1</code>	
<code>? 2 4</code>	

Задача С. Окна

Имя входного файла: windows.in
Имя выходного файла: windows.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

На экране расположены прямоугольные окна, каким-то образом перекрывающиеся (со сторонами, параллельными осям координат). Вам необходимо найти точку, которая покрыта наибольшим числом из них.

Формат входного файла

В первой строке входного файла записано число окон n ($1 \leq n \leq 50000$). Следующие n строк содержат координаты окон $x_{(1,i)}$ $y_{(1,i)}$ $x_{(2,i)}$ $y_{(2,i)}$, где $(x_{(1,i)}, y_{(1,i)})$ — координаты левого верхнего угла i -го окна, а $(x_{(2,i)}, y_{(2,i)})$ — правого нижнего (на экране компьютера y растет сверху вниз, а x — слева направо). Все координаты — целые числа, по модулю не превосходящие $2 \cdot 10^5$.

Формат выходного файла

В первой строке выходного файла выведите максимальное число окон, покрывающих какую-либо из точек в данной конфигурации. Во второй строке выведите два целых числа, разделенные пробелом — координаты точки, покрытой максимальным числом окон. Окна считаются замкнутыми, т.е. покрывающими свои граничные точки.

Пример

windows.in	windows.out
2	2
0 0 3 3	3 2
1 1 4 4	

Задача D. Перестановки

Имя входного файла: permutation.in
Имя выходного файла: permutation.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вася выписал на доске в каком-то порядке все числа от 1 по N , каждое число ровно по одному разу. Количество чисел оказалось довольно большим, поэтому Вася не может окинуть взглядом все числа. Однако ему надо всё-таки представлять эту последовательность, поэтому он написал программу, которая

отвечает на вопрос — сколько среди чисел, стоящих на позициях с x по y , по величине лежат в интервале от k до l . Сделайте то же самое.

Формат входного файла

В первой строке лежит два натуральных числа — $1 \leq N \leq 100\,000$ — количество чисел, которые выписал Вася и $1 \leq M \leq 100\,000$ — количество вопросов, которые Вася хочет задать программе. Во второй строке дано N чисел — последовательность чисел, выписанных Васей. Далее в M строках находятся описания вопросов. Каждая строка содержит четыре целых числа $1 \leq x \leq y \leq N$ и $1 \leq k \leq l \leq N$.

Формат выходного файла

Выведите M строк, каждая должна содержать единственное число — ответ на Васин вопрос.

Пример

permutation.in	permutation.out
4 2	1
1 2 3 4	3
1 2 2 3	
1 3 1 3	

Задача E. Звезды

Имя входного файла: stars.in
Имя выходного файла: stars.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вася любит наблюдать за звездами. Но следить за всем небом сразу ему тяжело. Поэтому он наблюдает только за частью пространства, ограниченной кубом размером $n \times n \times n$. Этот куб поделен на маленькие кубики размером $1 \times 1 \times 1$. Во время его наблюдений могут происходить следующие события:

1. В каком-то кубике появляются или исчезают несколько звезд.
2. К нему может заглянуть его друг Петя и поинтересоваться, сколько видно звезд в части пространства, состоящей из нескольких кубиков.

Формат входного файла

Первая строка входного файла содержит натуральное число $1 \leq n \leq 128$. Координаты кубиков — целые числа от 0 до $n - 1$. Далее следуют записи о

происходивших событиях по одной в строке. В начале строки записано число m . Если m равно:

- 1, то за ним следуют 4 числа — x, y, z ($0 \leq x, y, z < N$) и k ($-20000 \leq k \leq 20000$) — координаты кубика и величина, на которую в нем изменилось количество видимых звезд;
- 2, то за ним следуют 6 чисел — $x_1, y_1, z_1, x_2, y_2, z_2$ ($0 \leq x_1 \leq x_2 < N$, $0 \leq y_1 \leq y_2 < N$, $0 \leq z_1 \leq z_2 < N$), которые означают, что Петя попросил подсчитать количество звезд в кубиках (x, y, z) из области: $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, $z_1 \leq z \leq z_2$;
- 3, то это означает, что Васе надоело наблюдать за звездами и отвечать на вопросы Пети. Эта запись встречается во входном файле только один раз и будет последней записью.

Количество записей во входном файле не больше 100 002.

Формат выходного файла

Для каждого Петиного вопроса выведите на отдельной строке одно число — искомое количество звезд.

Пример

stars.in	stars.out
2	0
2 1 1 1 1 1 1	1
1 0 0 0 1	4
1 0 1 0 3	2
2 0 0 0 0 0 0	
2 0 0 0 0 1 0	
1 0 1 0 -2	
2 0 0 0 1 1 1	
3	

Задачи посложнее

Задача W. Золотые рудники

Имя входного файла: mine.in
Имя выходного файла: mine.out
Ограничение по времени: 0.25 секунд
Ограничение по памяти: 64 мегабайта

Байтмен, один из заслуженных работников компании по добыче золота в Байтленде, собирается в этом году на пенсию. Начальство конмпании решило вознаградить его за заслуги перед отечеством. Байтмену разрешили присвоить себе прямоугольную часть земли, со сторонами s и w , параллельными осям координат, со всеми входящими туда рудниками. Положение участка он выбирает сам. Назовем стоимостью участка количество рудников, лежащих внутри него и на его границе. Ваша задача вычислить максимально возможную стоимость такого участка.

Формат входного файла

В первой строке текстового файла mine.in находятся два целых числа s и w , разделенных одиночным пробелом ($1 \leq s, w \leq 10000$). Они означают длины сторон, параллельных осям Ox и Oy соответственно. Во второй строке находится целое число N ($1 \leq N \leq 15000$), количество рудников. В последующих N строках находятся координаты рудников. В $i + 2$ строке содержатся 2 целых числа x, y ($-30000 \leq x, y \leq 30000$), означающих координаты i -го рудника.

Формат выходного файла

Текстовый файл mine.out должен содержать одно число — ответ на задачу.

Пример

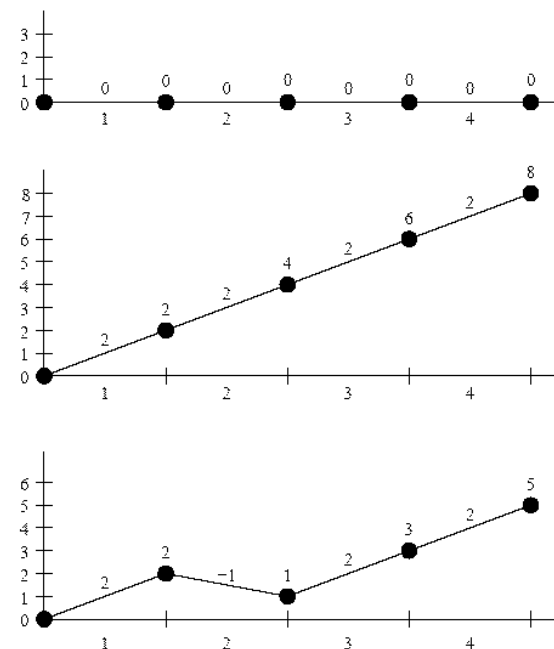
mine.in	mine.out
1 2	4
12	
0 0	
1 1	
2 2	
3 3	
4 5	
5 5	
4 2	
1 4	
0 5	
5 0	
2 3	
3 2	

Задача X. Горы

Имя входного файла: mountains.in
 Имя выходного файла: mountains.out
 Ограничение по времени: 2 секунды
 Ограничение по памяти: 64 мегабайта

В Горном Парке Развлечений открылся новый аттракцион с американскими горками. Трек в аттракционе состоит из n рельсов, соединённых последовательно друг с другом, причем первый рельс начинается на высоте 0. Оператор Байтмэн может изменять конфигурацию трека по своему усмотрению, корректируя наклон некоторых последовательно соединённых рельсов. Наклон остальных рельсов при этом не изменяется. Всякий раз, когда наклон некоторых рельсов изменяется, все следующие за ними рельсы соответственно поднимаются или опускаются. При этом высота начала трека всегда остается равной 0.

На рисунках представлены изменения конфигурации трека в соответствии со входными данными, заданными в примере:



Каждый заезд начинается с того, что кабина запускается из начала трека с энергией, достаточной для достижения высоты h . Иными словами, кабина будет продолжать движение по рельсам, пока ее высота не превысит h или пока она не достигнет конца трека. Вычислите по записям о всех заездах и изменениях конфигураций трека число рельсов, которые были полностью пройдены кабиной в каждом из заездов до ее остановки.

В аттракционе трек описывается последовательностью из n наклонов, по одному для каждого рельса. i -е число d_i равно разнице высот (в сантиметрах) между концом i -го рельса и его началом. Иными словами, если после прохождения по первым $i - 1$ рельсам кабина оказывается на высоте h сантиметров, то после прохождения по i -м рельсам она будет на высоте $h + d_i$ сантиметров. Изначально все рельсы горизонтальны, то есть $d_i = 0$ для всех i . Заезды и изменения конфигурации происходят в течение дня. Каждое изменение конфигурации описывается тремя числами: a , b и D . Такое изменение затрагивает рельсы с a -го по b -й (включительно). Наклон каждого из этих рельсов устанавливается равным D . Иными словами, $d_i = D$ для всех $a \leq i \leq b$. Каждый заезд задается одним

числом h — максимальной высотой, на которую может подняться кабина.

Напишите программу, которая:

- читает из входного файла последовательность описаний изменений конфигурации трека и заездов,
- для каждого заезда вычисляет количество рельсов, которые пройдены кабиной,
- выводит результаты в выходной файл.

Формат входного файла

Первая строка входного файла содержит одно положительное целое число n — количество рельсов в треке, $1 \leq n \leq 10^9$. Последующие строки содержат описания изменений конфигурации и заездов. Последняя строка входных данных содержит признак окончания. Каждая строка, начиная со второй, может содержать:

- Описание изменения конфигурации трека — один символ «I» и целые числа a, b и D , разделенные одним пробелом ($1 \leq a \leq b \leq n, -10^9 \leq D \leq 10^9$).
- Описание заезда — один символ «Q» и целое число h ($0 \leq h \leq 10^9$), разделённые одним пробелом.
- Один символ «E» — признак окончания входных данных.

Вы можете предполагать, что в каждый момент высота любой точки трека содержится в промежутке $[0, 10^9]$. Входной файл содержит не более 10^5 строк.

Формат выходного файла

i -я строка выходного файла должна содержать единственное целое число — количество рельсов, которые пройдены кабиной в i -м заезде.

Примеры

mountains.in	mountains.out
4	4
Q 1	1
I 1 4 2	0
Q 3	3
Q 1	
I 2 2 -1	
Q 3	
E	

Задача Y. Прямоугольники-online

Имя входного файла: `rect_online.in`
Имя выходного файла: `rect_online.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Вася нарисовал на плоскости N различных точек с целыми координатами. Теперь он хочет уметь быстро отвечать, а сколько точек лежит внутри заданного прямоугольника со сторонами, параллельными осям координат.

Формат входного файла

В первой строке задается количество точек N ($1 \leq N \leq 50\,000$). Последующие N строк содержат координаты точек $x_i y_i$. Следующая строка содержит число запросов M ($1 \leq M \leq 50\,000$). Следующие M строк содержат описания запросов, каждый в формате x_1, y_1, x_2, y_2 ($0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$).

Для каждого запроса нужно сделать две вещи:

- Посчитать число точек в соответствующем прямоугольнике (включая границы).
- Все посчитанные точки стереть.

Формат выходного файла

Для каждого из запросов выведите посчитанное количество точек.

Пример

rect_online.in	rect_online.out
9	4
0 0	5
0 1	0
0 2	
1 0	
1 1	
1 2	
2 0	
2 1	
2 2	
3	
0 0 1 1	
0 0 2 2	
0 0 3 3	

Задача Z. Caves and Tunnels (Пещеры и туннели)

Имя входного файла: caves.in
Имя выходного файла: caves.out
Ограничение по времени: 5 seconds
Ограничение по памяти: 64 megabytes

After landing on Mars surface, scientists found a strange system of caves connected by tunnels. So they began to research it using remote controlled robots. It was found out that there exists exactly one route between every pair of caves. But then scientists faced a particular problem. Sometimes in the caves faint explosions happen. They cause emission of radioactive isotopes and increase radiation level in the cave. Unfortunately robots don't stand radiation well. But for the research purposes they must travel from one cave to another. So scientists placed sensors in every cave to monitor radiation level in the caves. And now every time they move robots they want to know the maximal radiation level the robot will have to face during its relocation. So they asked you to write a program that will solve their problem.

Формат входного файла

The first line of the input contains one integer N ($1 \leq N \leq 100\,000$) — the number of caves. Next $N - 1$ lines describe tunnels. Each of these lines contains a pair of integers a_i, b_i ($1 \leq a_i, b_i \leq N$) specifying the numbers of the caves connected by corresponding tunnel. The next line has an integer Q ($Q \leq 100\,000$) representing the number of queries. The Q queries follow on a single line each. Every query has a form of " $C U V$ ", where C is a single character and can be either 'I' or 'G' representing the type of the query (quotes for clarity only). In the case of an 'I' query radiation level in U th cave ($1 \leq U \leq N$) is incremented by V ($0 \leq V \leq 10\,000$). In the case of a 'G' query your program must output the maximal level of radiation on the way between caves with numbers U and V ($1 \leq U, V \leq N$) after all increases of radiation ('I' queries) specified before current query. It is assumed that initially radiation level is 0 in all caves, and it never decreases with time (because isotopes' half-life time is much larger than the time of observations).

Первая строка входного файла содержит одно целое число N ($1 \leq N \leq 100\,000$) — количество пещер. Следующие $N - 1$ строк описывают туннели. Каждая из этих строк содержит два целых числа — a_i и b_i ($1 \leq a_i, b_i \leq N$), описывающие туннель из пещеры с номером a_i в пещеру с номером b_i . Следующая строка содержит целое число Q ($1 \leq Q \leq 100\,000$), означающее количество запросов. Далее идут Q запросов, по одному на строку. Каждый запрос имеет вид « $C U V$ », где C — символ «I» либо «G», означающие тип запроса (кавычки только для ясности). В случае запроса «I» уровень

радиации в U -й пещере ($1 \leq U \leq N$) увеличивается на V ($0 \leq V \leq 10\,000$). В случае запроса «G» ваша программа должна вывести максимальный уровень радиации на пути между пещерами с номерами U и V ($1 \leq U, V \leq N$) после всех увеличений радиации (запросов «I»), указанных ранее. Предполагается, что изначальный уровень радиации равен 0 во всех пещерах, и он никогда не уменьшается со временем (потому что период полураспада изотопов много больше времени наблюдения).

Формат выходного файла

For every 'G' query output one line containing the maximal radiation level by itself.

Для каждого запроса «G» выведите одну строку, содержащую максимальный уровень радиации.

Пример

caves.in	caves.out
4	1
1 2	0
2 3	1
2 4	3
6	
I 1 1	
G 1 1	
G 3 4	
I 2 3	
G 1 1	
G 3 4	