

Содержание

Распродажа баллов	2
Задача -2А. Ещё одна игра с камнями [0.1, 256]	2
Задача -2В. Дерево и отрезки [0.2, 256]	3
Задача -2С. Опекуны карнотавров [0.6, 256]	5
Задача -2D. Count Online [1.5, 256]	7
Задача -2Е. Конкатенация скобочных последовательностей [0.1, 256]	8
Задача -2F. Intercity Express [0.8, 256]	9
Задача -2G. Учимся красить [7.0, 256]	11
Задача -2H. Ближайшая большая справа [0.2, 256]	12
Задача -2I. Хорошие дни [0.1, 256]	13
Задача -2J. Yet another data structure [0.6, 256]	14
Задача -2К. Авиатуризм [1.0, 256]	15

У вас не получается читать/выводить данные?

Воспользуйтесь примерами (c++) (python).

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Распродажа баллов

Задача -2А. Ещё одна игра с камнями [0.1, 256]

Вася и Петя играют в неинтересную игру с самыми обычными правилами: на столе лежит сороконожка, у которой есть N ножек, а также камень. Первый игрок берёт со стола камень и отдавливая сороконожке одну ножку. Каждый следующий ход производится так: если на последнем ходу игрок отдал k ножек, то его соперник может отдать k или $k + 1$ ножку. Отдавленную один раз ножку нельзя отдать во второй раз. Проигрывает тот, кто не может ходить, то есть сороконожка. Однако нас интересует, выиграет первый игрок, или второй.

Формат входных данных

Входной файл содержит несколько тестов. Каждый тест представляет собой одну строку, на которой записано единственное целое число N ($1 \leq N \leq 100\,000$). Набор тестов завершается строкой с числом 0, которое не нужно обрабатывать.

Формат выходных данных

Для каждого теста выведите, кто выигрывает при правильной игре. Следуйте формату вывода, приведённому в примере, как можно точнее.

Пример

stdin	stdout
1	Case #1: First player wins.
2	Case #2: Second player wins.
3	Case #3: Second player wins.
0	

Подсказка по решению

Обычная динамика. $N^{3/2}$.

Задача -2В. Дерево и отрезки [0.2, 256]

Дерево отрезков — структура данных, позволяющая быстро изменять значения в массиве и находить некоторые функции от элементов $a[i], a[i + 1], \dots, a[j]$ массива.

Википедия

Филипп — фантазёр. На лекции ему рассказали о дереве отрезков, а он пришёл домой и придумал структуру данных «Дерево и отрезки».

С фантазией у Фили всё хорошо, а вот с программированием — не очень. Поэтому реализовать «Дерево и отрезки» предстоит Вам.

Вам даётся дерево, каждая вершина имеет свой цвет. Затем даётся несколько отрезков. Необходимо для каждой вершины дерева узнать, сколько вершин в её поддереве (включая её саму) имеют цвет, номер которого принадлежит этому отрезку.

Формат входных данных

Все числа в файле целые.

$0 \leq N \leq 3 \cdot 10^5$, $0 \leq Q \leq 10$ — количество вершин и запросов.

Затем N пар чисел $1 \leq p_i, c_i \leq N$ — i -ая вершина имеет родителя с номером p_i и покрашена в цвет c_i . Для корня дерева $p_i = 0$.

Следующие Q строк — запросы. Каждый запрос — пара чисел l_i, r_i ($1 \leq l_i \leq r_i \leq N$).

Формат выходных данных

Вывести N строк по Q чисел, обозначающих количества цветов в поддеревьях с корнями в вершинах $1, \dots, N$ в соответствующих отрезках.

Примеры

stdin	stdout
5 2 2 1 3 2 0 3 3 3 2 1 1 5 2 3	1 0 3 1 5 3 1 1 1 0
1 1 0 1 1 1	1
2 1 0 1 1 1 1 1	2 1
2 3 0 1 1 2 1 1 2 2 1 2	1 1 2 0 1 1

Подсказка по решению

Очень простая задача. Не перемудрите.

Задача -2С. Опекуны карнотавров [0.6, 256]

Карнотавры очень внимательно относятся к заботе о своем потомстве. У каждого динозавра обязательно есть старший динозавр, который его опекает. В случае, если опекуна съедают (к сожалению, в юрский период такое не было редкостью), забота о его подопечных ложится на плечи того, кто опекал съеденного динамозавра. Карнотавры — смертоносные хищники, поэтому их обычаи строго запрещают им драться между собой. Если у них возникает какой-то конфликт, то, чтобы решить его, они обращаются к кому-то из старших, которому доверяют, а доверяют они только тем, кто является их опекуном или опекуном их опекуна и так далее (назовем таких динозавров суперопекунами). Поэтому для того, чтобы решить спор двух карнотавров, нужно найти такого динозавра, который является суперопекуном для них обоих. Разумеется, беспокоить старших по пустякам не стоит, поэтому спорщики стараются найти самого младшего из динозавров, который удовлетворяет этому условию. Если у динозавра возник конфликт с его суперопекуном, то этот суперопекун сам решит проблему. Если у динозавра нелады с самим собой, он должен разобраться с этим самостоятельно, не беспокоя старших. Помогите динозаврам разрешить их споры.

Формат входных данных

Во входном файле записано число M , обозначающее количество запросов ($1 \leq M \leq 200\,000$). Далее на отдельных строках следуют M запросов, обозначающих следующие события:

- $+ v$ — родился новый динозавр и опекунство над ним взял динозавр с номером v . Родившемуся динозавру нужно присвоить наименьший натуральный номер, который до этого еще никогда не встречался.
- $- v$ — динозавра номер v съели.
- $? u v$ — у динозавров с номерами u и v возник конфликт и вам надо найти им третейского судью.

Изначально есть один прадинозавр номер 1; гарантируется, что он никогда не будет съеден.

Формат выходных данных

Для каждого запроса типа «?» в выходной файл нужно вывести на отдельной строке одно число — номер самого молодого динозавра, который может выступить в роли третейского судьи.

Примеры

stdin	stdout
11	1
+ 1	1
+ 1	2
+ 2	2
? 2 3	5
? 1 3	
? 2 4	
+ 4	
+ 4	
- 4	
? 5 6	
? 5 5	

Подсказка по решению

Опять какие-то деревья, опять какие-то подъёмы...

Задача -2D. Count Online [1.5, 256]

Вам дано множество точек на плоскости.

Нужно уметь отвечать на два типа запросов:

○ ? $x_1 y_1 x_2 y_2$ — сказать, сколько точек лежит в прямоугольнике $[x_1..x_2] \times [y_1..y_2]$. Точки на границе и в углах тоже считаются. $x_1 \leq x_2, y_1 \leq y_2$.

○ + $x y$ — добавить в множество точку $(x + \text{res} \% 100, y + \text{res} \% 101)$. Где res — ответ на последний запрос вида ?, а $\%$ — операция взятия по модулю.

Формат входных данных

Число точек N ($1 \leq N \leq 50\,000$). Далее N точек.

Число запросов Q ($1 \leq Q \leq 100\,000$). Далее Q запросов. Все координаты от 0 до 10^9 .

Формат выходных данных

Для каждого запроса GET одно целое число — количество точек внутри прямоугольника.

Пример

stdin	stdout
5	3
0 0	3
1 0	1
0 1	0
1 1	0
1 1	3
9	
? 0 1 1 2	
+ 1 2	
+ 2 2	
? 1 0 2 2	
? 0 0 0 0	
+ 3 3	
? 3 3 3 3	
? 4 3 4 3	
? 4 4 5 5	

Подсказка по решению

На самом деле добавлялись точки $(4, 5), (5, 5), (4, 4)$.

Корневая по запросам, а внутри уже готовое решение?

Задача -2Е. Конкатенация скобочных последовательностей [0.1, 256]

Даны n скобочных последовательностей из круглых скобок. Выбрать некоторое подмножество, выписать в произвольном порядке так, чтобы

- Результат являлся правильной скобочной последовательностью.
- Длина результата была максимально возможной.

Формат входных данных

На первой строке n ($1 \leq n \leq 1000$). Следующие n строк содержат непустые скобочные последовательности, суммарная длина не более 10 000.

Формат выходных данных

В первой строке выведите через пробел числа l и k , где l — максимальная длина правильной скобочной последовательности, а k — количество кусков, из которых она состоит. Во второй строке через пробел выведите k номеров использованных кусков в том порядке, в котором их следует выписывать. Куски занумерованы числами от 1 до n в том порядке, в котором они даны на входе. Если возможных ответов несколько, выведите любой.

Примеры

stdin	stdout
4 ((((())	4 3 1 3 4
3 ((()	6 3 2 3 1

Подсказка по решению

Жадность? Или динамика? Что-то похожее у нас было.

Задача -2F. Intercity Express [0.8, 256]

Андрей разрабатывает систему для продажи железнодорожных билетов. Он собирается протестировать ее на Междугородней Экспресс линии, которая соединяет два больших города и имеет $n - 2$ промежуточных станций, то есть в итоге есть n станций, пронумерованных от 1 до n .

В Междугороднем Экспресс поезде есть s мест, пронумерованных с 1 до s . В тестирующем режиме система имеет доступ к базе данных, содержащей проданные билеты в направлении от станции 1 до станции n и должна отвечать на вопросы, можно ли продать билет от станции a до станции b , и если да, нужно найти минимальный номер места, которое свободно на протяжении всего пути между a и b .

Изначально система имеет только доступ на чтение, то есть даже если есть свободное место, она должна сообщить об этом, но не должна изменять данные.

Помогите Андрею протестировать его систему написанием программы, которые будет находить ответы на вопросы.

Формат входных данных

Первая строка содержит число n — количество станций, s — количество мест и m — количество уже проданных билетов ($2 \leq n \leq 10^9$, $1 \leq s \leq 100\,000$, $0 \leq m \leq 100\,000$).

В следующих m строках описаны билеты, описание каждого билета состоит из трех чисел: c_i , a_i и b_i — номер места, которое занимает владелец билета, номер станции, с которой продан билет и номер станции, до которой продан билет ($1 \leq c_i \leq s$, $1 \leq a_i < b_i \leq n$).

Следующие строки содержат число q — количество запросов ($1 \leq q \leq 100\,000$). Специальное значение p должно поддерживаться в течение считывания запросов. Изначально $p = 0$.

Следующие $2q$ строк описывают запросы. Каждый запрос описывается двумя числами: x_i и y_i ($x_i \leq y_i$).

Чтобы получить города a и b между которыми нужно проверить наличие места, используется следующая формула:

$a = x_i + p$, $b = y_i + p$. Ответ на запрос — число 0, если нет места на каждом отрезке между a и b , или минимальный номер свободного места.

После ответа на запрос, надо приравнять число p полученному ответу на запрос.

Формат выходных данных

Для каждого запроса выведите ответ на него.

Примеры

stdin	stdout
5 3 5	1
1 2 5	2
2 1 2	2
2 4 5	3
3 2 3	0
3 3 4	2
10	0
1 2	0
1 2	0
1 2	0
2 3	0
-2 0	
2 4	
1 3	
1 4	
2 5	
1 5	

Подсказка по решению

Обратите внимание, что запросы выглядят так: (1, 2), (2, 3), (3, 4), (4, 5), (1, 3), (2, 4), (3, 5), (1, 4), (2, 5), (1, 5).

Можно думать персистентно. А можно корневой.

Задача -2G. Учимся красить [7.0, 256]

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — покрасить его вершины в минимальное количество цветов таким образом, чтобы смежные вершины были покрашены в разные цвета.

Формат входных данных

На первой строке число вершин $n \geq 1$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Формат выходных данных

На первой строке выведите k — минимальное количество цветов. На следующей строке n целых чисел от 1 до k — цвета вершин. Если оптимальных раскрасок несколько, выведите любую.

Система оценки

$n \leq 70$ (возможно, в итоге уменьшится до $n \leq 60$).

Примеры

stdin	stdout
5 8	4
5 4	1 1 2 3 4
3 5	
1 5	
1 3	
2 3	
1 4	
5 2	
3 4	

Подсказка по решению

Random Walk. Мелкие степени. Жадность. Random. Непростая задача.

Задача -2Н. Ближайшая большая справа [0.2, 256]

Дан массив a из n чисел. Нужно обрабатывать запросы:

0. `set(i, x)` – $a[i] = x$;
1. `get(i, x)` – найти $\min k: k \geq i$ и $a_k \geq x$.

Формат входных данных

На первой строке длина массива n и количество запросов m . На второй строке n целых чисел – массив a . Следующие m строк содержат запросы.

Индексы в массиве нумеруются с 1.

Запрос типа `set`: “0 i x”.

Запрос типа `get`: “1 i x”.

$$1 \leq n, m \leq 200\,000.$$

$$1 \leq i \leq n.$$

$$0 \leq x, a_i \leq 200\,000.$$

Формат выходных данных

На каждой запрос типа `get` на отдельной строке выведите k .

Если такого k не существует, выведите -1 .

Примеры

stdin	stdout
4 5	1
1 2 3 4	3
1 1 1	-1
1 1 3	2
1 1 5	
0 2 3	
1 1 3	

Подсказка по решению

Научились на практике. Типичное Д.О.

Задача -2I. Хорошие дни [0.1, 256]

Билл разрабатывает новую математическую теорию, описывающую человеческие эмоции. Его последние исследования посвящены изучению того, насколько хорошие и плохие дни влияют на воспоминания людей о различных периодах жизни.

Недавно Билл придумал методику, которая описывает, насколько хорошим или плохим был день человеческой жизни с помощью сопоставления дню некоторого неотрицательного целого числа. Билл называет это число *эмоциональной значимостью* этого дня. Чем больше это число, тем лучше этот день. Билл полагает, что значимость некоторого периода человеческой жизни равна сумме эмоциональных значимостей каждого из дней периода, помноженной на минимум эмоциональных значимостей дней этого периода. Эта методика отражает то, что период, который в среднем может быть весьма неплох, бывает испорчен одним плохим днем.

Теперь Билл хочет проанализировать свою собственную жизнь и найти в ней период максимальной значимости. Помогите ему это сделать.

Формат входных данных

Первая строка входного файла содержит число n — количество дней в жизни Билла, которые он хочет исследовать ($1 \leq n \leq 100\,000$). Оставшаяся часть файла содержит n целых чисел a_1, a_2, \dots, a_n , все в пределах от 0 до 10^6 — эмоциональные значимости дней. Числа во входном файле разделяются пробелами и переводами строки.

Формат выходных данных

В первой строке выходного файла выведите максимальную значимость периода жизни Билла. Во второй строке выведите два числа l и r , означающие, что значимость периода с l -го по r -й день (включительно) в жизни Билла была максимально возможной.

Примеры

stdin	stdout
6	60
3 1 6 4 5 2	3 5

Подсказка по решению

Просто задача на структуры данных.

Задача -2J. Yet another data structure [0.6, 256]

Нужно отвечать на запросы вида

- + x – добавить в мультимножество число x .
- ? x – посчитать сумму чисел не больших x .

Формат входных данных

Каждая строка файла содержит один запрос.

Все числа x целые от 0 до $10^9 - 1$.

Количество запросов от 1 до 10^5 .

Формат выходных данных

Ответы на все запросы вида “? x ”.

Примеры

stdin	stdout
+ 1	3
+ 2	3
? 3	1
? 2	0
? 1	2
? 0	
+ 1	
? 1	

Подсказка по решению

Здесь не обязательно писать оптимально. Зайдёт всё, что угодно.

Задача -2К. Авиатуризм [1.0, 256]

Авиационная сеть Байтландии устроена следующим образом. В каждом из n городов, занумерованных натуральными числами от 1 до n , построен аэропорт. В стране работают несколько авиакомпаний, причём i -я авиакомпания обслуживает двунаправленные авиарейсы, соединяющие каждый из городов с номерами в интервале $[a_i, b_i]$ с каждым из городов с номерами в интервале $[c_i, d_i]$. Полёт на каждом авиарейсе стоит один байтландский тугрик. При этом между любыми двумя городами Байтландии существует авиационный маршрут (возможно, с пересадками).

Администрация города с номером k издаёт туристический справочник, в котором, в частности, должны быть приведена таблица минимальной стоимости авиаперелёта между городом k и каждым из городов Байтландии. Разработку этой таблицы администрация поручила Вам.

Формат входных данных

В первой строке входного файла заданы три целых числа n , m и k ($2 \leq n \leq 500\,000, 1 \leq m \leq 100\,000, 1 \leq k \leq n$) - количество городов в Байтландии, количество авиакомпаний и номер города, для которого издаётся справочник.

Последующие m строк содержат описания рейсов каждой компании, по одному на строку. Описание каждого рейса состоит из четырёх целых чисел a_i, b_i, c_i, d_i ($1 \leq a_i \leq b_i \leq n, 1 \leq c_i \leq d_i \leq n$, пересечение $[a_i, b_i]$ и $[c_i, d_i]$ пусто), обозначающих, что каждый из городов с номерами a_i, \dots, b_i соединён двунаправленной авиалинией с каждым из городов с номерами c_i, \dots, d_i . При этом возможна ситуация, что между двумя городами прямые рейсы выполняет более одной авиакомпании.

Формат выходных данных

Выведите n строк, i -я из которых содержит одно число - минимальную стоимость авиаперелёта из города с номером k в город с номером i (в частности, в k -й строке должно быть число 0).

stdin	stdout
5 3 4	1
1 2 4 5	1
5 5 4 4	2
1 1 3 3	0
	1

Подсказка по решению

bfs и... что же добавить к bfs?