

Содержание

Обязательные задачи	3
Задача 5A. Корень кубического уравнения [0.2 sec, 256 mb]	3
Задача 5B. Коровы – в стойла! [0.2 sec, 256 mb]	4
Задача 5C. Минимальный максимум [1.8 sec, 256 mb]	5
Задача 5D. Одномерный почтальон [0.8 sec, 256 mb]	6
Задача 5E. Одномерный финансист [0.2 sec, 256 mb]	7
Задача 5F. Точки и отрезки [0.3 sec, 256 mb]	8
Задача 5G. Объединение прямоугольников [0.5 sec, 256 mb]	9
Задача 5H. La cucaracha [0.2 sec, 256 mb]	10
Дополнительные задачи	11
Задача 5I. Добро пожаловать на сборы [2.5 sec, 256 mb]	11
Задача 5J. Возня на круге [3 sec, 256 mb]	12
Задача 5K. Сортировка за линейное время [0.6 sec, 256 mb]	13
Задача 5L. Инверсии отрезка [0.7 sec, 256 mb]	14
Задача 5M. И снова сумма... [3 sec, 256 mb]	15

Во всех задачах можно использовать `std::sort`, `std::set`.

Во всех задачах **нельзя** использовать дерево отрезков, декартово дерево.

Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html

Выделение памяти.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Обязательные задачи

Задача 5А. Корень кубического уравнения [0.2 sec, 256 mb]

Дано кубическое уравнение $ax^3 + bx^2 + cx + d = 0$ ($a \neq 0$). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

Формат входных данных

Во входном файле через пробел записаны четыре целых числа: $-1000 \leq a, b, c, d \leq 1000$.

Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 6 знаков после десятичной точки.

Пример

cubroot.in	cubroot.out
1 -3 3 -1	1
-1 -6 -12 -7	-1.000000

Задача 5В. Коровы – в стойла! [0.2 sec, 256 mb]

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное расстояние между коровами было как можно больше.

Формат входных данных

В первой строке вводятся числа N ($2 < N < 10001$) — количество стойл и K ($1 < K < N$) — количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания — координаты стойл (координаты не превосходят 10^9).

Формат выходных данных

Выведите одно число — наибольшее возможное допустимое расстояние.

Примеры

cows.in	cows.out
5 3 1 2 3 100 1000	99

Задача 5С. Минимальный максимум [1.8 сек, 256 mb]

Даны n нестрого возрастающих массивов A_i и m нестрого убывающих массивов B_j . Все массивы имеют одну и ту же длину l . Далее даны q запросов вида (i, j) , ответ на запрос – такое k , что $\max(A_{ik}, B_{jk})$ минимален. Если таких k несколько, можно вернуть любое.

Формат входных данных

На первой строке числа n, m, l ($1 \leq n, m \leq 900; 1 \leq l \leq 3000$). Следующие n строк содержат описания массивов A_i . Каждый массив описывается перечислением l элементов. Элементы массива – целые числа от 0 до $10^5 - 1$. Далее число m и описание массивов B_j в таком же формате. Массивы и элементы внутри массива нумеруются с 1. На следующей строке число запросов q ($1 \leq q \leq n \cdot m$). Следующие q строк содержат пары чисел i, j ($1 \leq i \leq n, 1 \leq j \leq m$).

Формат выходных данных

Выведите q чисел от 1 до l – ответы на запросы.

Примеры

minmax.in	minmax.out
4 3 5	3
1 2 3 4 5	4
1 1 1 1 1	3
0 99999 99999 99999 99999	5
0 0 0 0 99999	4
5 4 3 2 1	3
99999 99999 99999 0 0	1
99999 99999 0 0 0	1
12	1
1 1	4
1 2	4
1 3	3
2 1	
2 2	
2 3	
3 1	
3 2	
3 3	
4 1	
4 2	
4 3	

Задача 5D. Одномерный почтальон [0.8 сек, 256 mb]

В деревне Печалька живут n человек, их домики расположены ровно на оси абсцисс. Домик i -го человека находится в точке x_i . В деревню приехал и хочет там поселиться почтальон. Координату своего домика y он хочет выбрать так, чтобы суммарное расстояние от него до всех жителей деревни было минимально возможным. То есть

$$\sum_{i=1}^n |y - x_i| \rightarrow \min$$

Вам дан массив x из n случайных целых чисел. Найдите точку y .

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^7$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до 224 - 1.
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до 232 - 1.
9. }
```

Элементы массива генерируются последовательно. $x_i = \text{nextRand32}()$;

Формат выходных данных

Выведите одно число — минимальное суммарное расстояние от точки y до всех домиков.

Примеры

postman.in	postman.out
6 239 13	8510257371

Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.

Задача 5Е. Одномерный финансист [0.2 сек, 256 mb]

В деревне Печалька живут n человек, их домики расположены ровно на оси абсцисс. Домик i -го человека находится в точке x_i . В деревню недавно заселился почтальон. Почтальон построил себе домик в такой точке y , что суммарное расстояние от него до всех жителей деревни было минимально возможным. А теперь в деревню приехал финансовый аналитик, который привык не только оптимизировать результат, но и оценивать риски. Посмотрев на опыт почтальона, аналитик заметил, что несмотря на то, что сумма минимальна, есть домики очень далеко от дома почтальона. Финансист учел это и свой дом хочет построить в такой точке z , что

$$\sum_{i=1}^n (z - x_i)^2 \rightarrow \min$$

С почтальоном финансист не дружит, поэтому расстояние до y в сумме не учитывается. Вам дан массив x из n случайных целых чисел. Найдите точку z .

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^7$). На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до  $2^{32} - 1$ .
9. }
```

Элементы массива генерируются последовательно. $x_i = \text{nextRand32}()$;

Формат выходных данных

Выведите координату домика финансиста в виде несократимой дроби с положительным знаменателем.

Примеры

finansist.in	finansist.out
6	3368129374/3
230 10	

Замечание

Сгенерированный массив: 9, 1004452, 2338007883, 149525792, 917993446, 3329727166.

Задача 5F. Точки и отрезки [0.3 sec, 256 mb]

На прямой в точках $a_1, a_2, \dots, a_n$ (возможно, совпадающих) сидят n котят. На той же прямой лежат m отрезков $[l_1, r_1], [l_2, r_2], \dots, [l_m, r_m]$. Нужно для каждого отрезка узнать его *наполненность котятми* — сколько котят сидит на отрезке.

Формат входных данных

На первой строке n и m ($1 \leq n, m \leq 10^5$). На второй строке n целых чисел a_i ($0 \leq a_i \leq 10^9$). Следующие m строк содержат пары целых чисел l_i, r_i ($0 \leq l_i \leq r_i \leq 10^9$).

Формат выходных данных

Выведите m целых чисел. i -е число — *наполненность котятми* i -го отрезка.

Примеры

segpnt.in	segpnt.out
4 5 1 2 3 5 1 4 1 5 2 5 3 3 0 0	3 4 3 1 0

Задача 5G. Объединение прямоугольников [0.5 sec, 256 mb]

На плоскости задано N прямоугольников с вершинами в точках с целыми координатами и сторонами, параллельными осям координат. Необходимо найти площадь их объединения.

Формат входных данных

В первой строке входного файла указано число N ($0 \leq N \leq 1500$). В следующих N строках заданы по 4 целых числа x_1, y_1, x_2, y_2 — сначала координаты левого нижнего угла прямоугольника, потом правого верхнего ($0 \leq x_1 \leq x_2 \leq 10^9, 0 \leq y_1 \leq y_2 \leq 10^9$). Обратите внимание, что прямоугольники могут вырождаться в отрезки и даже в точки.

Формат выходных данных

В выходной файл выведите единственное число — ответ на задачу.

Пример

union.in	union.out
3 1 1 3 5 5 2 7 4 2 4 6 7	23

Задача 5Н. La cucaracha [0.2 sec, 256 mb]

Каждую полночь в квартире ученого Васи начинается ужас. Сотни . . . , о нет! ТЫСЯЧИ тараканов вылазят из каждой дырки к его обеденному столу, уничтожая все крошки и объедки! Вася ненавидит тараканов. Он очень долго думал и сделал Супер-ловушку, которая привлекает всех тараканов в большой зоне после активации. Он планирует активировать ловушку сегодня ночью. Но есть проблема. Эта очень эффективная ловушка с её очень большой зоной работы поглощает огромное количество энергии. Так что, Вася планирует минимизировать время работы этой ловушки. Он собрал информацию о всех местах, в которых живут тараканы. Также он заметил, что все тараканы двигаются только по линиям его скатерти с постоянной скоростью (мы можем предположить, что эта скорость равна 1, так что таракан расположенный в одной из секций, может за 1 единицу времени переместиться на любую соседнюю секцию (по вертикали или горизонтали)). Вася решил активировать его ловушку в одной из секций. Когда ловушка активирована, все тараканы будут двигаться к секции, содержащей ловушку, так быстро, как только смогут. Поэтому в любой момент времени после активации тараканы двигаются к секции, в которой находится ловушка, максимально уменьшая расстояние до неё. Если есть два пути с одинаковым расстоянием, то таракан выберет любой. Напишите программу для Васи, которая выбирает секцию, минимизирующую время, необходимое для уничтожения всех тараканов. Конечно, ваша программа будет считать, что скатерть будет плоскостью с декартовой системой координат и секции — точки с целыми координатами.

Формат входных данных

В первой строке входного файла содержится число мест, в которых живут тараканы N ($1 \leq N \leq 10000$). Следующие N строк содержат x и y — координаты мест, в которых живут тараканы (целые числа не больше 10^9 по абсолютному значению).

Формат выходных данных

Вам необходимо вывести только два целых числа x и y , не превосходящие по модулю 10^9 , — координаты секции, которая минимизирует время работы. Если есть более одно решение — выведите любое из них.

Пример

cucarach.in	cucarach.out
2	2 2
1 1	
3 3	

Дополнительные задачи

Задача 51. Добро пожаловать на сборы [2.5 sec, 256 mb]

Как известно, всенародные сборы по программированию и информированию собирают участников со всей страны Б.! Поэтому организаторы сборов решили в следующий раз выбрать место проведения таким образом, чтобы минимизировать суммарное время, которое потребуется всем участникам для того, чтобы добраться до выбранного места.

Для простоты будем считать, что страна Б. представляет собой плоскость со стандартной декартовой системой координат, участники изначально расположены в целочисленных точках этой плоскости, из точки (x, y) он может перейти в точку (x_1, y_1) за время $\max(|x - x_1|, |y - y_1|)$. Необходимо выбрать точку с рациональными координатами для места проведения сборов так, чтобы минимизировать **суммарное** время, потраченное участниками на дорогу до места сборов.

Вам даны n групп участников. i -я группа расположена в точке x_i, y_i и имеет численность w_i . Найдите минимальное суммарное время в виде несократимой дроби.

Формат входных данных

В первой строке записано единственное число t ($1 \leq t \leq 10^5$) — количество тестовых примеров. Каждый тестовый пример описывается следующим образом:

В первой строке описания записано целое число n — количество участников сборов. В следующих n строках описываются места, в которых изначально расположены участники. Каждое описание состоит из трёх целых чисел x, y ($0 \leq x, y \leq 10^9$) и w ($1 \leq w \leq 10^9$) и говорит о том, что в точке с координатами (x, y) изначально расположены w участников. Гарантируется, что сумма всех w в каждом тестовом примере не превосходит 10^9 .

Сумма всех n во входных данных не превосходит 3 000 000.

Формат выходных данных

В первой строке выведите число t — количество тестовых примеров, в каждой из следующих t строк выведите ответ на соответствующий тест — числитель и знаменатель несократимой дроби.

Пример

nearest.in	nearest.out
2	2
4	66 1
3 1 10	4 1
7 0 9	
5 6 8	
4 3 1	
3	
1 1 1	
2 2 2	
3 3 3	

Задача 5J. Возня на круге [3 сек, 256 mb]

Представьте себе клетчатую окружность, состоящую из L клеток. Клетки нумеруются целыми числами от 1 до L . Некоторые N клеток закрашены. Окружность можно разрезать между любыми двумя клетками. Всего существует L различных разрезов. Получившаяся полоска делится на K равных частей (L кратно K). $\forall i$ определим f_i , как количество закрашенных клеток в i -й части. Вам нужно найти такой разрез, что

$$F = \max_{i=1..K} f_i - \min_{i=1..K} f_i$$

минимально возможно. Кроме того, вам нужно найти количество разрезов, на которых достигается максимум.

Формат входных данных

На первой строке целые числа L , N , K , ($K \geq 2$). Гарантируется, что L делится на K . На второй строке N различных целых чисел от 1 до L — номера закрашенных клеток.

Формат выходных данных

На первой строке выведите минимальное значение F и количество разрезов с таким F . На второй строке выведите любое x от 1 до L , что, если разрезать окружность между клетками x и $x + 1$, а после чего посчитать F , то получится максимальное значение.

Ограничения: $N \leq 500\,000$, $L \leq 10^{18}$.

Примеры

circle.in	circle.out
10 1 5 7	1 10 3
10 4 2 3 7 5 2	0 4 3

Пояснение к примеру

В первом примере подойдет любой разрез.

Во втором примере разрезать можно после 3, 4, 8, 9 клеток.

Задача 5К. Сортировка за линейное время [0.6 сек, 256 mb]

Дан массив случайных целых чисел, нужно отсортировать его.

Формат входных данных

На первой строке количество тестов t ($1 \leq t \leq 200$) и число n ($1 \leq n \leq 50\,000$) — размер массива в каждом из тестов. На второй строке пара целых чисел a, b от 1 до 10^9 , используемая в генераторе случайных чисел.

```
1. unsigned int cur = 0; // беззнаковое 32-битное число
2. unsigned int nextRand24() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur >> 8; // число от 0 до  $2^{24} - 1$ .
5. }
6. unsigned int nextRand32() {
7.     unsigned int a = nextRand24(), b = nextRand24();
8.     return (a << 8) ^ b; // число от 0 до  $2^{32} - 1$ .
9. }
```

Тесты генерируются последовательно.

Элементы массива генерируются последовательно. $x_i = \text{nextRand32}()$;

Формат выходных данных

Для каждого теста выведите на отдельной строке $\sum_{i=1}^n x_i \cdot i$.

Примеры

buckets.in	buckets.out
1 6 239 13	46062181379

Замечание

Сгенерированный массив: 12, 130926, 3941054950, 2013898548, 197852696, 2753287507.

В этой задаче очень небольшой запас по времени. Если она не сдается, это нормально.

Задача 5L. Инверсии отрезка [0.7 sec, 256 mb]

Дана перестановка p из n чисел. Ваша задача — быстро обрабатывать запросы вида *количество инверсий на отрезке данной перестановки*. Напомним, инверсией называется пара индексов (i, j) : $i < j$, $p_i > p_j$.

Формат входных данных

На первой строке целое число n .

На второй строке сама перестановка.

На третьей строке количество запросов q .

Следующие q строк содержат пары целых чисел l_i, r_i ($1 \leq l_i \leq r_i \leq n$).

Формат выходных данных

Для каждого из q запросов выведите ответ на отдельной строке.

Система оценки

Подзадача 1 (50 баллов) $1 \leq n, q \leq 10^5$ $l_i \leq l_{i+1}, r_i \leq r_{i+1}$.

Подзадача 2 (50 баллов) $1 \leq n, q \leq 20\,000$ l_i и r_i произвольны

Пример

invseg.in	invseg.out
5	3
5 4 3 1 2	9
3	2
1 3	
1 5	
3 5	

Замечание

В этой задаче можно пользоваться чем угодно. Даже деревом отрезков.

Задача 5М. И снова сумма... [3 sec, 256 mb]

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 входных данных

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 $add((i + y) \bmod 10^9)$.

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

Формат выходных данных

Для каждого запроса выведите одно число — ответ на запрос.

Пример

sum2.in	sum2.out
6	3
+ 1	7
+ 3	
+ 3	
? 2 4	
+ 1	
? 2 4	

Замечание

Это задача на “пополняемый массив”.