Содержание

Общая информация:

Вход в контест: `http://contest.yandex.ru/contest/758/`
Дедлайн на задачи 27-го октябрбя в 23:59.
К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).
В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.
Сайт курса: `http://compscicenter.ru/courses/algorithms-1/2014-autumn/`
Семинары ведет Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии 2 таймлимита: для `C/C++` и для `Java, Python`.

03.Base [3/3]

1 Задача 5A. Сумма простая [3 sec (4.5 sec)]

Вам нужно научиться отвечать на запрос “сумма чисел на отрезке”.

Массив не меняется. Запросов много. Отвечать на 1 запрос следует за $O(1)$.

Формат входных данных

Размер массива — n и числа x, y, a_0, порождающие массив a: $a_i = (x \cdot a_{i-1} + y) \bmod 2^{16}$ Далее следуют количество запросов m и числа z, t, b_0, порождающие массив b: $b_i = (z \cdot b_{i-1} + t) \bmod 2^{30}$, $c_i = b_i \bmod n$. i-й запрос — найти сумму на отрезке от $min(c_{2i}, c_{2i+1})$ до $max(c_{2i}, c_{2i+1})$ в массиве a.

Ограничения: $1 \leqslant n \leqslant 10^7, 0 \leqslant m \leqslant 10^7$. Все числа целые от 0 до 2^{16}. t может быть -1.

Формат выходных данных

Выведите сумму всех сумм.

Пример

sum0.in	sum0.out
3 1 2 3 3 1 -1 4	23

Замечание

$a = \{3, 5, 7\}, b = \{4, 3, 2, 1, 0, 2^{30} - 1\}, c = \{1, 0, 2, 1, 0, 0\}$,
запросы $= \{[0, 1], [1, 2], [0, 0]\}$, суммы $= \{8, 12, 3\}$.

2 Задача 5B. Быстрое прибавление [6 sec (9 sec)]

Есть массив целых чисел длины $n = 2^{24}$, изначально заполненных нулями. Вам нужно сперва обработать m случайных запросов вида "прибавление на отрезке" по модулю 2^{32}. Затем обработать q случайных запросов вида "сумма на отрезке" по модулю 2^{32}.

Замечание

Это очень простая задача. Не нужны никакие деревья.

Если у вас есть запрос `[l..r] += x`, достаточно сделать `a[l] += x, a[r+1] -= x;`.

Формат входных данных

На первой строке числа m, q. ($1 \leqslant m, q \leqslant 2^{24}$). 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 nextRand() {
3.     cur = cur * a + b; // вычисляется с переполнениями
4.     return cur » 8; // число от 0 до 2^24 − 1.
5. }
```

Каждый запрос первого вида генерируется следующим образом:

```
1. add = nextRand(); // число, которое нужно прибавить
2. l = nextRand();
3. r = nextRand();
4. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Каждый запрос второго вида генерируется следующим образом:

```
1. l = nextRand();
2. r = nextRand();
3. if (l > r) swap(l, r); // получили отрезок [l..r]
```

Сперва генерируются запросы первого вида, затем второго.

Формат выходных данных

Выведите сумму ответов на все запросы по модулю 2^{32}.

Примеры

fastadd.in	fastadd.out
5 5 13 239	811747796

3 Задача 5C. Разрезание графа [0.5 sec (1 sec)]

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

Формат входных данных

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа n, количество рёбер m и количество операций k ($1 \leqslant n \leqslant 50\,000$, $0 \leqslant m \leqslant 100\,000$, $m \leqslant k \leqslant 150\,000$).

Следующие m строк задают рёбра графа; i-ая из этих строк содержит два числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$), разделённые пробелами — номера концов i-го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют k строк, описывающих операции. Операция типа `cut` задаётся строкой "`cut` u v" ($1 \leqslant u, v \leqslant n$), которая означает, что из графа удаляют ребро между вершинами u и v. Операция типа `ask` задаётся строкой "`ask` u v" ($1 \leqslant u, v \leqslant n$), которая означает, что необходимо узнать, лежат ли в данный момент вершины u и v в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

Формат выходных данных

Для каждой операции `ask` во входном файле выведите на отдельной строке слово "`YES`", если две указанные вершины лежат в одной компоненте связности, и "`NO`" в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

Пример

cutting.in	cutting.out
3 3 7 1 2 2 3 3 1 ask 3 3 cut 1 2 ask 1 2 cut 1 3 ask 2 1 cut 2 3 ask 3 1	YES YES NO NO

Замечание

Подсказка: попробуйте обрабатывать запросы с конца.

03.Advanced [2/6]

4 Задача 5D. RMQ [1 sec (2 sec)]

Дан массив `a[1..n]`. Требуется написать программу, обрабатывающую два типа запросов.

- Запрос "`max` l r". Требуется найти максимум в массиве `a` от l-ой ячейки до r-ой включительно.
- Запрос "`add` l r v". Требуется прибавить значение v к каждой ячейке массива `a` от l-ой до r-ой включительно.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \le n, q \le 10^5$) – длина массива и число запросов соответственно. Второая строка содержит n целых чисел $a_1, \ldots, a_n$ ($|a_i| \le 10^5$), задающих соответствующие значения массива. Следующие q строк содержат запросы.

В зависимости от типа запрос может иметь вид либо "`max` l r", либо "`add` l r v". При этом $1 \le l \le r \le n$, $|v| \le 10^5$.

Формат выходных данных

Для каждого запроса вида "`max` l r" требуется в отдельной строке выдать значение соответствующего максимума.

Примеры

rmq.in	rmq.out
5 3 1 2 3 4 -5 max 1 3 add 1 2 5 max 1 3	3 7

5 Задача 5E. Сумма [0.8 sec (1.6 sec)]

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

Формат входных данных

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

1. A l r x — присвоить элементам массива с позициями от l до r значение x ($1 \leqslant l \leqslant r \leqslant N$, $0 \leqslant x \leqslant 10^9$)
2. Q l r — найти сумму чисел в массиве на позициях от l до r. ($1 \leqslant l \leqslant r \leqslant N$)

Изначально массив заполнен нулями.

Формат выходных данных

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

Примеры

sum.in	sum.out
5 9 A 2 3 2 A 3 5 1 A 4 5 2 Q 1 3 Q 2 2 Q 3 4 Q 4 5 Q 5 5 Q 1 5	3 2 3 4 2 7

6 Задача 5F. Все минимумы [2 sec (4 sec)]

Дан массивы целых чисел $a_1, a_2, \ldots, a_n$.

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

Найдите

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

то есть сумму минимумов всех подотрезков.

Внимание. Есть очень простое решение за $O(n)$. Ваше решение может быть любым, главное чтобы укладывалось в TL.

Формат входных данных

Первая строка входных данных содержит натуральное число n ($1 \leqslant n \leqslant 100\,000$) — 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

minsum.in	minsum.out
1 5	5
2 -10 1	-19
4 1 2 3 4	20
5 -3 2 -4 1 -5	-52

7 Задача 5G. Хорошие дни [1 sec (2 sec)]

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 ($1 \leqslant n \leqslant 100\,000$). Оставшаяся часть файла содержит n целых чисел $a_1, a_2, \ldots, a_n$, 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

feelgood.in	feelgood.out
6 3 1 6 4 5 2	60 3 5

8 Задача 5H. Художник [1.2 sec (2.4 sec)]

Итальянский художник-абстракционист Ф. Мандарино увлекся рисованием одномерных черно-белых картин. Он пытается найти оптимальное местоположение и количество черных участков картины. Для этого он проводит на прямой белые и черные отрезки, и после каждой из таких операций хочет знать количество черных отрезков на получившейся картине и их суммарную длину.

Изначально прямая — белая. Ваша задача — написать программу, которая после каждой из таких операций выводит в выходной файл интересующие художника данные.

Формат входных данных

В первой строке входного файла содержится общее количество нарисованных отрезков ($1 \leqslant N \leqslant 100\,000$). В последующих N строках содержится описание операций. Каждая операция описывается строкой вида $c\ x\ l$, где c — цвет отрезка (`W` для белых отрезков, `B` для черных), а сам отрезок имеет вид $[x; x+l]$, причем координаты обоих концов — целые числа, не превосходящие по модулю 500 000. Длина задается положительным целым числом.

Формат выходных данных

После выполнения каждой из операций необходимо вывести в выходной файл на отдельной строке количество черных отрезков на картине и их суммарную длину, разделенные одним пробелом.

Пример

painter.in	painter.out
7 W 2 3 B 2 2 B 4 2 B 3 2 B 7 2 W 3 1 W 0 10	0 0 1 2 1 4 1 4 2 6 3 5 0 0

9 Задача 5I. Обратная инверсия-2 [2 sec (4 sec)]

Таблицей инверсий для перестановки $A = (a_1, a_2, \ldots, a_n)$ чисел $\{1, 2, \ldots, N\}$ называется массив $X = (x_i)_{1 \le i \le N}$, в котором на i-м месте стоит количество элементов, больших i, но стоящих левее, чем i, т.е $x_i =$ число таких j', что$j' < j, a_{j'} > a_j = i$.

Например, таблицей инверсий для перестановки $(2, 5, 1, 3, 4)$ будет $(2, 0, 1, 1, 0)$, а для перестановки $(6, 1, 3, 7, 5, 4, 2)$ — $(1, 5, 1, 3, 2, 0, 0)$.

Обратной перестановкой A^{-1} к перестановке A называется такая перестановка чисел, что на i-м месте в A^{-1} стоит номер места, на котором стоит элемент, равный i, в перестановке A.

Например, для перестановки $(2, 5, 1, 3, 4)$ обратной будет $(3, 1, 4, 5, 2)$ (т.к. 1 стоит на третьем месте, 2 — на первом, 3 — на четвертом, 4 — на пятом, а 5 — на втором), а для перестановки $(2, 7, 3, 6, 5, 1, 4)$ обратной будет $(6, 1, 3, 7, 5, 4, 2)$.

Ваша задача — по таблице инверсий перестановки A посчитать таблицу инверсий обратной перестановки A^{-1}.

Формат входных данных

Файл состоит ровно из N чисел, разделенных пробелами и переводами строки, задающих таблицу инверсий перестановки A. Число N находится в пределах от 1 до **262 144**.

Формат выходных данных

Выведите N целых чисел, разделенных пробелами — таблицу инверсий для обратной перестановки.

Пример

invers2.in	invers2.out
2 0 1 1 0	1 3 0 0 0
5 0 1 3 2 1 0	1 5 1 3 2 0 0

03.Hard [0/1]

10 Задача 5J. Сортировка за линейное время [0.7 sec (impossible)]

Дан массив случайных целых чисел, нужно отсортировать его.

Формат входных данных

На первой строке количество тестов t ($1 \leqslant t \leqslant 200$) и число n ($1 \leqslant n \leqslant 50\,000$) — 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 = `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.