

Содержание

Задачи здорового человека	2
Задача 11А. Словарная Задача [0.3 sec, 256 mb]	2
Задача 11В. Обмен [0.4 sec, 256 mb]	3
Для искателей острых ощущений	4
Задача 11С1. Сила с тобой, Люк [6 sec, 256 mb]	4
Задача 11С2. Сила с тобой, Люк (+) [1.6 sec, 256 mb]	5

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

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

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

Обратите внимание на GNU C++ компиляторы с суффиксом `inc`, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Задачи здорового человека

Задача 11А. Словарная Задача [0.3 sec, 256 mb]

Нужно обрабатывать запросы двух типов: добавить число x в мультимножество, удалить число x из мультимножества.

Формат входных данных

Количество запросов n ($1 \leq n \leq 100\,000$).

Следующие n строк содержат запросы двух типов «+ x » и «- x ».

Все числа целые от 1 до 10^9 .

Формат выходных данных

Если после очередного запроса «- x » чисел x в мультимножестве больше не осталось, выведите «after query i number x disappeared», где i — номер запроса. Если при очередном запросе «- x » числа x уже нет, выведите «query $\#i$: can not delete x ».

Примеры

stdin	stdout
8	query #1: can not delete 7
- 7	after query 5 number 7 disappeared
+ 7	after query 7 number 2 disappeared
+ 7	query #8: can not delete 2
- 7	
- 7	
+ 2	
- 2	
- 2	

Подсказка по решению

```
map<int,int>? unordered_map<int,int>?
```

Задача 11В. Обмен [0.4 sec, 256 mb]

Пусть все натуральные числа исходно организованы в список в естественном порядке. Разрешается выполнить следующую операцию: $swap(a, b)$. Эта операция возвращает в качестве результата расстояние в текущем списке между числами a и b и меняет их местами.

Задана последовательность операций $swap$. Требуется вывести в выходной файл результат всех этих операций.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 200\,000$) — количество операций. Каждая из следующих n строк содержит по два числа в диапазоне от 1 до 10^9 — аргументы операций $swap$.

Формат выходных данных

Для каждой операции во входном файле выведите ее результат.

Пример

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

Замечание

Хеш-таблицы из STL (`unordered_map` и т.д.) получают TL.

Придётся писать свою хеш-таблицу.

Самая быстрая и простая хеш-таблица – хеш-таблица с открытой адресацией.

Если вы получаете TL 17 или 18, у вас плохая хеш-функция.

Для искателей острых ощущений

Задача 11C1. Сила с тобой, Люк [6 sec, 256 mb]

Дан массив a из n чисел, нужно научиться обрабатывать запросы двух типов.

- `change(i, y)` — сделать a_i равным y .
- `intget()` — вернуть индекс i такой, что a_i встречается в массиве наименьшее возможное число раз. Если таких i несколько, вернуть минимально возможный.

Все индексы, встречающиеся в задаче, нумеруются с нуля.

Формат входных данных

На первой строке размер массива n .

На второй строке массив — n целых чисел от 0 до 10^9-1 .

На третьей строке число запросов q .

Следующие q строк содержат запросы в формате «?» (`get`) и «= i y » (`change`).

Ограничения: $n, q \leq 300\,000$.

Формат выходных данных

На каждый запрос `get` выведите на отдельной строке ответ.

Примеры

stdin	stdout
4	0
1 2 3 4	2
5	0
?	
= 0 2	
?	
= 3 3	
?	
6	2
1 1 2 1 1 0	2
3	
?	
= 5 2	
?	

Замечание

Это задача про STL!

Мы проходим структуры данных. Очень важно уметь пользоваться стандартными.

В C++: STL есть много полезного. Вот напоминание того, чем уже пора уметь пользоваться.

<http://acm.math.spbu.ru/~sk1/examples/c++/stl/sample06-data-structures.cpp.html>

По этой задаче можно получить ОК за 0.750 sec, используя только контейнеры C++: STL.

Подсказка по решению

Поскольку 8.5 мегабайт нельзя прочитать из файла мгновенно (как и записать 1 мегабайт данных), используйте максимально быстрые ввод вывод.

Задача 11C2. Сила с тобой, Люк (+) [1.6 сек, 256 mb]

Ровно та же задача, но TL пожёстче.