

Задача А. Fixed set (75 баллов)

Ограничение по времени: 1.5 секунд
Ограничение по памяти: 64Mb

Реализуйте следующий класс для хранения множества целых чисел:

```
class FixedSet {  
public:  
    FixedSet();  
    void Initialize(const vector<int>& numbers);  
    bool Contains(int number) const;  
};
```

`FixedSet` получает при вызове `Initialize` набор целых чисел, который впоследствии и будет хранить. Набор чисел не будет изменяться с течением времени (до следующего вызова `Initialize`). Операция `Contains` возвращает `true`, если число `number` содержится в наборе. Мат. ожидание времени работы `Initialize` должно составлять $O(n)$, где n — количество чисел в `numbers`. Затраты памяти должны быть порядка $O(n)$ в худшем случае. Операция `Contains` должна выполняться за $O(1)$ в худшем случае.

С помощью этого класса решите модельную задачу: во входе будет дано множество различных чисел, а затем множество запросов — целых чисел. Необходимо для каждого запроса определить, лежит ли число из запроса в множестве.

Формат входных данных

В этой задаче ввод, вывод и `main` уже написаны за вас. Вы можете найти их здесь: <https://gist.github.com/astiunov/810433327e6b4c9f6ebd60ddb2b00430>. Та же программа запускается на сервере. Вы должны прислать заголовочный файл с реализацией класса `FixedSet`. Обратите внимание, что в программе есть небольшое количество юнит-тестов, которые также запускаются на сервере. Ваше решение должно их проходить, помимо тестов в самом контексте. Тесты можно запустить, передав флаг `--testing` вашей программе.

В задаче строится множество из n различных чисел, после чего выполняется q запросов. $1 \leq n \leq 100\,000$, $1 \leq q \leq 1\,000\,000$, все числа в задаче по модулю не превосходят 10^9 .

Примеры

тест	ответ
3	Yes
1 2 3	Yes
4	Yes
1 2 3 4	No
3	No
3 1 2	Yes
4	No
10 1 5 2	Yes

Задача В. Два королевства (85 баллов)

Ограничение по времени: 1 секунда
Ограничение по памяти: 42Mb

Нынешним утром Боб хвастался перед Джо тем, что на днях побывал в совершенно удивительном королевстве X . Слушая его рассказ, Джо вдруг подумал, что королевство, о котором говорит Боб очень похоже на то, в котором побывал сам Джо лет эдак 50 назад. С этим путешествием у Джо связаны приятные воспоминания, и даже сохранилась старинная карта этого королевства Y , на которой отмечены все города.

Итак, на этой карте отмечено N городов, каждый из которых имеет натуральный номер от 1 до N . Между некоторыми городами построены дороги, по которым можно двигаться в обе стороны. Королевство устроено так, что от каждого города до любого другого можно добраться единственным способом, возможно, заходя на пути в некоторые другие города.

Боб, как настоящий путешественник, посетил все города королевства X , а также составил полную его карту. Удивительное дело, но в этом королевстве также N городов, пронумерованных числами от 1 до N . По карте Боба видно, что королевство X тоже устроено таким образом, что от каждого города можно добраться до любого другого единственным способом, возможно, заходя на пути в некоторые другие города.

За долгие 50 лет нумерация городов совершенно изменилась. Но Джо будет бесконечно вам благодарен, если вы определите, действительно ли Боб побывал в том же королевстве, что и Джо когда-то? И если это так, то он хочет знать для каждого города его новый номер.

Формат входных данных

В первой строке записано натуральное число N ($1 \leq N \leq 10^5$) — количество городов в королевстве. В следующих $N - 1$ строках дано описание королевства Y , а именно пары чисел: номера городов, соединенных дорогой. Следующие $N - 1$ строк описывают королевство X в том же формате.

Формат выходных данных

Если Джо ошибся, и королевства разные, то выведите «-1» (без кавычек). Иначе выведите N чисел, каждое в отдельной строке. i -ое число должно быть равно новому номеру города i . Если решений несколько, выведите любое.

Пример

тест	ответ
4	1
3 1	2
2 3	4
4 3	3
2 4	
4 1	
3 4	

Задача С. Менеджер памяти (75 баллов)

Ограничение по времени: 0.5 секунд
Ограничение по памяти: 64Mb

Пете поручили написать менеджер памяти для новой стандартной библиотеки языка C++. В распоряжении у менеджера находится массив из N последовательных ячеек памяти, пронумерованных от 1 до N . Задача менеджера — обрабатывать запросы приложений на выделение и освобождение памяти. Запрос на выделение памяти имеет один параметр K . Такой запрос означает, что приложение просит выделить ему K последовательных ячеек памяти. Если в распоряжении менеджера есть хотя бы один свободный блок из K последовательных ячеек, то он обязан в ответ на запрос выделить такой блок. При этом наш менеджер выделяет память из самого длинного свободного блока, а если таких несколько, то из них он выбирает тот, у которого номер первой ячейки — наименьший. После этого выделенные ячейки становятся занятыми и не могут быть использованы для выделения памяти, пока не будут освобождены. Если блока из K последовательных свободных ячеек нет, то запрос отклоняется. Запрос на освобождение памяти имеет один параметр T . Такой запрос означает, что менеджер должен освободить память, выделенную ранее при обработке запроса с порядковым номером T . Запросы нумеруются, начиная с единицы. Гарантируется, что запрос с номером T — запрос на выделение, причем к нему еще не применялось освобождение памяти. Освобожденные ячейки могут снова быть использованы для выделения памяти. Если запрос с номером T был отклонен, то текущий запрос на освобождение памяти игнорируется. Требуется написать симуляцию менеджера памяти, удовлетворяющую приведенным критериям.

Формат входных данных

В первой строке входа два числа N и M — количество ячеек памяти и запросов соответственно ($1 \leq N \leq 2^{31} - 1$, $1 \leq M \leq 10^5$). Каждая из следующих M строк содержит по одному числу. $(i + 1)$ -я строка содержит положительное число K , если i -й запрос — запрос на выделение K ячеек памяти ($1 \leq K \leq N$), и отрицательное число $-T$, если i -й запрос — запрос на освобождение памяти, выделенной по запросу номер T ($1 \leq T < i$).

Формат выходных данных

Для каждого запроса на выделение памяти выведите в выход одно число на отдельной строке с результатом выполнения этого запроса. Если память была выделена, выведите номер первой ячейки памяти в выделенном блоке, иначе выведите число -1 .

Пример

тест	ответ
6 8	1
2	3
3	-1
-1	-1
3	1
3	-1
-5	
2	
2	

Задача D. О скользящей k -й статистике (75 баллов)

Ограничение по времени: 1 секунда
Ограничение по памяти: 64Mb

По массиву двигаются два указателя — l и r , которые изначально направлены на самый первый элемент массива. Есть две операции: L — сдвинуть указатель l на один элемент вправо и R — сдвинуть указатель r на один элемент вправо. После выполнения каждой из операций нужно вывести k -е в порядке возрастания число среди всех чисел от l до r включительно, либо -1 , если всего чисел от l до r меньше, чем k .

Формат входных данных

В первой строке входа — три целых числа n, m, k . Во второй строке n целых чисел, задающих массив чисел, по которому будут двигаться два указателя, l и r . В третьей строке входного файла — m символов R или L , без пробелов, в одну строку. Это порядок выполняемых операций. Гарантируется, что указатель l никогда не «обгоняет» указатель r . Гарантируется, что указатель r никогда не выйдет за пределы массива. При этом $1 \leq n, k \leq 100\,000$, $0 \leq m \leq 2n - 2$. Все числа в массиве неотрицательные и не превосходят 10^9 .

Формат выходных данных

Выведите ровно m строк, в каждой — ровно по одному целому числу. После выполнения каждой из операций нужно вывести k -е в порядке возрастания число среди всех чисел от l до r включительно, либо -1 , если всего чисел от l до r меньше, чем k .

Примеры

тест	ответ
7 4 2 4 2 1 3 6 5 7 RRLL	4 2 2 -1
4 6 1 1 2 3 4 RLRLRL	1 2 2 3 3 4

Задача Е. Классы подобия треугольников (55 баллов)

Ограничение по времени: 2 секунды

Ограничение по памяти: 256Мб

Вам дан набор из n треугольников ($1 \leq n \leq 1\,000\,000$). Каждый треугольник задается тремя целыми числами — длинами его сторон. Длины сторон — положительные целые числа, не превосходящие 10^9 . Некоторые пары треугольников подобны. В силу свойств подобия известно, что все эти треугольники распадаются на несколько классов, в каждом из которых все треугольники подобны друг другу. К примеру, три треугольника со сторонами $(6, 6, 10)$, $(15, 25, 15)$, $(35, 21, 21)$ все попарно подобны друг другу, то есть образуют один класс. Необходимо найти количество классов подобия, на которые распадаются данные n треугольников.

Сложность алгоритма должна в среднем линейно зависеть от n (может еще зависеть от ограничений на стороны треугольника). Т.е. никаких сортировок длинных массивов в решении делать нельзя.

Формат входных данных

В первой строке входа дано число n . В каждой из следующих n строк — по три целых числа, задающих стороны очередного треугольника.

Формат выходных данных

Выведите число классов подобия.

Примеры

тест	ответ
3 6 6 10 15 25 15 35 21 21	1
4 3 4 5 10 11 12 6 7 8 6 8 10	3