Средняя

**Дерево отрезков с операцией на отрезке**

**Ограничение по времени работы: 1 секунда**

Реализуйте эффективную структуру данных для хранения элементов и увеличения нескольких подряд идущих элементов на одно и то же число.

ВХОДНЫЕ ДАННЫЕ

Первая строка входных данных содержит одно натуральное число N (1⩽N⩽100000) — количество элементов в массива. Во второй строке записаны N чисел от 0 до 100000 — элементы массива.

В третьей строке вводится одно натуральное число M (1⩽M⩽30000) — количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (g — получить текущее значение элемента по его номеру, a — увеличить все элементы на отрезке).

Следом за командой g вводится одно число — номер элемента (от 1 до N).

Следом за a вводится три числа — левый и правый конец отрезка и значение add, на которое нужно увеличить все элементы данного отрезка (1⩽add⩽100000).

ВЫХОДНЫЕ ДАННЫЕ

Программа должна вывести ответы на все запросы g.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 2 4 3 5 2 5 g 2 g 5 a 1 3 10 g 2 g 4 |  |

Сложная

**Билеты на электричку**

**Ограничение по времени работы: 2 секунды**

В новых электричках каждому пассажиру положено сидячее место. Естественно, количество сидячих мест ограничено и на всех их может не хватить. Маршрут электрички проходит через N станций, занумерованных от 0 до N−1. Когда человек хочет купить билет, он называет два числа x и y – номера станций, откуда и куда он хочет ехать. При наличии хотя бы одного сидячего места на этом участке на момент покупки ему продается билет, иначе выдается сообщение «билетов нет» и билет не продается. Номера мест не учитываются, важно только наличие хотя бы одного свободного места на каждом участке пути. Ваша задача – написать программу, обслуживающую такого рода запросы в порядке их прихода.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных содержатся три числа N – количество станций (2⩽N⩽200000), K – количество мест в электричке (1⩽K⩽1000) и M – количество запросов (1⩽M⩽100000). В следующих M строках описаны запросы, каждый из которых состоит из двух чисел x и y (0⩽x<y<N).

ВЫХОДНЫЕ ДАННЫЕ

На каждый запрос ваша программа должна выдавать результат в виде числа 0, если билет не продается, и 1, если билет был продан. Каждый результат должен быть на отдельной строке

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 2 4  0 4  1 2  1 4  2 4 | 1  1  0  1 |

Сложная

**Непересекающиеся отрезки**

Отрезок целочисленной прямой длины N разбит на единичные отрезки, которые пронумерованы от 1 до N. Их объединяют в группы по следующим правилам: 1. Несколько подряд идущих отрезков, ни один из которых не принадлежит ни одной из групп, могут быть объединены в группу.  
2. Любая ранее созданная группа может быть уничтожена, при этом входившие в нее отрезки больше не относятся ни к какой группе и могут впоследствии быть отнесены к другим группам. Видно, что любой отрезок всегда находится не более, чем в одной группе. Каждую группу можно идентифицировать парой чисел: номером первого и номером последнего отрезка, входящего в группу. Первоначально нет ни одной группы.

ВХОДНЫЕ ДАННЫЕ

Первая строка входных данных содержит число N — количество отрезков и число K — количество запросов (1≤N,K≤105). Далее идет K строчек, содержащих запросы к структуре данных. Каждый запрос начинается с числа 1 (запрос на создание группы) или 2 (запрос на удаление группы). После числа 1 указывается два других числа l и  r (1≤l≤r≤N), после числа 2 указывается одно число  i (1≤i≤N).

ВЫХОДНЫЕ ДАННЫЕ

Для каждого запроса типа 1 необходимо отрезки с номерами от l до r объединить в группу. Если все эти отрезки не входят ни в одну группу, запрос считается удачным и программа должна вывести 1. Если хотя бы один из этих отрезков уже относится к какой-то группе, запрос считается неудачным, объединение не производится и программа выводит 0. Для каждого запроса типа 2 необходимо удалить группу, в которую входит отрезок с номером i, при этом программа должна вывести два числа: номер первого и последнего отрезка, входящих в удаляемую группу. Если отрезок с номером i не относится ни к одной группе, программа должна вывести два нуля.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 6  1 1 2  1 4 5  1 2 4  2 5  2 1  2 4 | 1  1  0  4 5  1 2  0 0 |

Сложная

**Zeroes**

Злобный учитель в Фоксфорде любит мучить детей сложными задачками. А если дети эти задачки не решают, учитель подвергает их самым жестоким наказаниям. На этот раз он придумал такую задачу: Рейтинг всех учеников Фоксфорда записан в массив A Запросы учителя таковы: 1. Изменить рейтинг i-го ученика на число x  
2. Найти максимальную последовательность подряд идущих нулей (бесперспективных учеников) в массиве A на отрезке [l,r].  
Помогите бедным ученикам Фоксфорда избежать зверского наказания за нерешение задачи на этот раз.

ВХОДНЫЕ ДАННЫЕ

В первой строке входного файла записано число N(1≤N≤500000) – количество учеников в Фоксфорде. Во второй строке записано N чисел – их рейтинги, числа по модулю не превосходящие 1000 (по количеству задач, которые ученик решил или не решил за время обучения). В третьей строке записано число M(1≤M≤50000) – количество запросов. Каждая из следующих M строк содержит описания запросов: UPDATE  i x – обновить i-ый элемент массива значением  x (1≤i≤N,|x|≤1000). QUERY  l r – найти длину максимальной последовательности из нулей на отрезке с l по  r. (1≤l≤r≤N).

ВЫХОДНЫЕ ДАННЫЕ

В выходной файл выведите ответы на запросы QUERY в том же порядке, что и во входном файле.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5  328 0 0 0 0  5  QUERY 1 3  UPDATE 2 832  QUERY 3 3  QUERY 2 3  UPDATE 2 0 | 2  1  1 |