

Задача А. Дерево отрезков с операцией на отрезке

Имя входного файла: `segment-tree.in`
Имя выходного файла: `segment-tree.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

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

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30\,000$) — 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 'g' вводится одно число — номер элемента.

Следом за 'a' вводится три числа — левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($0 \leq add \leq 100\,000$).

Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос 'g'.

Примеры

<code>segment-tree.in</code>	<code>segment-tree.out</code>
5	4
2 4 3 5 2	2
5	14
g 2	5
g 5	
a 1 3 10	
g 2	
g 4	

Задача В. Поиск максимума

Имя входного файла: `index-max.in`
Имя выходного файла: `index-max.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Реализуйте структуру данных для эффективного вычисления номера максимального из нескольких подряд идущих элементов массива.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100\,000$) — количество чисел в массиве.

Во второй строке вводятся N чисел от 1 до 100 000 — элементы массива.

В третьей строке вводится одно натуральное число K ($1 \leq K \leq 30\,000$) — количество запросов на вычисление максимума.

В следующих K строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

Формат выходных данных

Для каждого запроса выведите индекс максимального элемента на указанном отрезке массива. Если максимальных элементов несколько, выведите любой их них.

Числа выводите в одну строку через пробел.

Примеры

<code>index-max.in</code>	<code>index-max.out</code>
5	3
2 2 2 1 5	5
2	
2 3	
2 5	

Замечание

TL для Python 3.5 секунды

Задача С. Прибавление и максимум

Имя входного файла: `addandmax.in`
Имя выходного файла: `addandmax.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Реализуйте эффективную структуру данных для хранения массива и выполнения следующих операций: увеличение всех элементов данного интервала на одно и то же число; поиск максимума на интервале.

Формат входных данных

В первой строке вводится одно натуральное число N ($1 \leq N \leq 100000$) – количество чисел в массиве.

Во второй строке вводятся N чисел от 0 до 100000 – элементы массива.

В третьей строке вводится одно натуральное число M ($1 \leq M \leq 30000$) – количество запросов.

Каждая из следующих M строк представляет собой описание запроса. Сначала вводится одна буква, кодирующая вид запроса (m – найти максимум на отрезке (оба конца включительно), a – увеличить все элементы на отрезке, то есть все элементы с l -го по r -й).

Следом за m вводятся два числа – левая и правая граница отрезка.

Следом за a вводятся три числа – левый и правый концы отрезка и число add , на которое нужно увеличить все элементы данного отрезка массива ($0 \leq add \leq 100000$).

Формат выходных данных

Выведите в одну строку через пробел ответы на каждый запрос m .

Примеры

<code>addandmax.in</code>	<code>addandmax.out</code>
5	4
2 4 3 1 5	104
5	104
m 1 3	
a 2 4 100	
m 1 3	
a 5 5 10	
m 1 5	

Задача D. Знакочередование

Имя входного файла: `signchange.in`
Имя выходного файла: `signchange.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Реализуйте структуру данных из n элементов a_1, a_2, \dots, a_n , поддерживающую следующие операции:

- присвоить элементу a_i значение j ;
- найти знакопередающуюся сумму на отрезке от l до r включительно, т. е. $(a_l - a_{l+1} + a_{l+2} - \dots - a_r)$.

Формат входных данных

В первой строке входного файла содержится натуральное число n ($1 \leq n \leq 10^5$) — длина массива. Во второй строке записаны начальные значения элементов — неотрицательные целые числа, не превосходящие 10^4 .

В третьей строке находится натуральное число m ($1 \leq m \leq 10^5$) — количество операций. В последующих m строках записаны операции:

- операция первого типа задаётся тремя числами $0 \ i \ j$ ($1 \leq i \leq n, 1 \leq j \leq 10^4$).
- операция второго типа задаётся тремя числами $1 \ l \ r$ ($1 \leq l \leq r \leq n$).

Формат выходных данных

Для каждой операции второго типа выведите на отдельной строке соответствующую знакопередающуюся сумму.

Пример

<code>signchange.in</code>	<code>signchange.out</code>
3	-1
1 2 3	2
5	-1
1 1 2	3
1 1 3	
1 2 3	
0 2 1	
1 1 3	

Задача Е. Разгрузочная задача для отдыха (на дереве)

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Мы решили, что пора переходить к более простым темам, а поэтому, чтобы переход был относительно плавный, эта задача по-тихому закралась в контекст на ДО.

Известны два натуральных числа a и b , с которыми происходит следующий процесс:

1. Все натуральные числа от a до b включительно перемножаются между собой (то есть получается число $\frac{b!}{(a-1)!}$)
2. У полученного числа берется сумма цифр
3. Если результат второго шага не меньше 10, процесс возвращается на второе шаг еще раз

Кому-то очень сильно понадобилось число, которое получается в конце этого процесса (можно доказать, что процесс всегда заканчивается). Помогите им найти результат процесса!

Формат входных данных

В первой строке дано число a , на второй — число b ($1 \leq a \leq b < 10^{18}$). Оба числа даны без ведущих нулей.

Формат выходных данных

Выведите число, которое получится в конце процесса.

Примеры

стандартный ввод	стандартный вывод
1 5	3
6 8	3