Задача Painter. Художник

Имя входного файла:	`painter.in`
Имя выходного файла:	`painter.out`
Ограничение по времени:	4 секунды
Ограничение по памяти:	64 мегабайта

Не успев дорисовать свой гениальный футуристический шедевр, М. Калевич 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 такой операции выводит в выходной файл интересующие художника данные.

Формат входного файла

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 выходного файла

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

Задача Sparse. Разреженные таблицы

Имя входного файла:	`sparse.in`
Имя выходного файла:	`sparse.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мебибайта

Дан массив из n чисел. Требуется написать программу, которая будет отвечать на запросы следующего вида: найти минимум на отрезке между u и v включительно.

Формат входного файла

В первой строке входного файла даны три натуральных числа n, m ($1 \leqslant n \leqslant 10^5, m \leqslant 10^7$) и a_1 ($0 \leqslant a_1 < 16714589$) — количество элементов в массиве, количество запросов и первый элемент массива соответственно. Вторая строка содержит два натуральных числа u_1 и v_1 ($1 \leqslant u_1, v_1 \leqslant n$) — первый запрос.

Элементы $a_2, a_3, \ldots, a_n$ задаются следующей формулой:

$$a_{i+1} = (23 \cdot a_i + 21563) \mod 16714589$$

Например, при $n = 10$, $a_1 = 12345$ получается следующий массив: a = (12345, 305498, 7048017, 11694653, 1565158, 2591019, 9471233, 570265, 13137658, 1325095).

Запросы генерируются следующим образом:

$$u_{i+1} = ((17 \cdot u_i + 751 + ans_i + 2i) \mod n) + 1$$
$$v_{i+1} = ((13 \cdot v_i + 593 + ans_i + 5i) \mod n) + 1$$

где ans_i — ответ на запрос номер i.

Формат выходного файла

В выходной файл выведите u_m, v_m и ans_m (последний запрос и ответ на него).

Пример

sparse.in	sparse.out
10 8 12345 3 9	5 3 1565158

Задача Tree. Декартово дерево

Имя входного файла:	`tree.in`
Имя выходного файла:	`tree.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Вам даны пары чисел (a_i, b_i), Вам необходимо построить декартово дерево, такое что i-ая вершина имеет ключи (a_i, b_i), вершины с ключем a_i образуют бинарное дерево поиска, а вершины с ключем b_i образуют кучу.

Формат входного файла

В первой строке записано число N — количество пар. Далее следует N ($1 \leqslant N \leqslant 50\,000$) пар (a_i, b_i). Для всех пар $|a_i|, |b_i| \leqslant 30\,000$. $a_i \neq a_j$ и $b_i \neq b_j$ для всех $i \neq j$.

Формат выходного файла

Если декартово дерево с таким набором ключей построить возможно, выведите в первой строке `YES`, в противном случае выведите `NO`. В случае ответа `YES`, выведите N строк, каждая из которых должна описывать вершину. Описание вершины состоит из трёх чисел: номер предка, номер левого сына и номер правого сына. Если у вершины отсутствует предок или какой-либо из сыновей, то выводите на его месте число `0`.

Если подходящих деревьев несколько, выведите любое.

Пример

tree.in	tree.out
7 5 4 2 2 3 9 0 5 1 3 6 6 4 11	YES 2 3 6 0 5 1 1 0 7 5 0 0 2 4 0 1 0 0 3 0 0