

Задача 8. Обходы бинарного дерева

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Бинарное дерево — это набор вершин, у каждой из которых может быть левый и правый ребёнок. Одна из вершин является корнем дерева, она не является ребёнком какой-то другой. Начав в корне и каждый раз переходя в одного из детей, можно прийти до любой вершины. Множество вершин, до которых можно прийти из заданной, называется её поддеревом.

У бинарного дерева есть три основных обхода: прямой (*pre-order*), центрированный (*in-order*) и обратный (*post-order*).

Прямой обход дерева — это порядок его вершин, полученный следующим рекурсивным алгоритмом:

1. Добавить корень дерева в обход.
2. Если у корня есть левый ребёнок, выписать прямой обход его поддерева.
3. Если у корня есть правый ребёнок, выписать прямой обход его поддерева.

В центрированном обходе корень дерева выписывается между обходами поддеревьев его детей, в обратном — после обходов поддеревьев его детей. Во всех вариантах обхода для каждой вершины сначала обходится левое поддерево, а затем правое.

Обобщим эти три варианта обхода: пусть в каждой вершине записано целое число x от -1 до 1 , обозначающее, в какой момент мы выписываем эту вершину, а именно:

- $x = -1$: до обходов поддеревьев её детей;
- $x = 0$: между обходами поддеревьев её детей;
- $x = 1$: после обходов поддеревьев её детей.

Таким образом, если во всех вершинах записано -1 , обход является прямым, если 0 — центрированным, если 1 — обратным.

Рассмотрим дерево с n вершинами, пронумерованных от 1 до n . Корень дерева — вершина 1 . Изначально во всех вершинах записано число -1 .

В рамках исследования необходимо обработать q запросов одного из следующих типов:

1. Поменять числа в вершинах $l, l + 1, \dots, r$ на x (x равен $-1, 0$ или 1).
2. Сообщить, на какой позиции в текущем обходе будет стоять вершина i .

Необходимо вывести ответы на все запросы второго типа.

Формат входных данных

В первой строке входных данных даны два целых числа n и q ($1 \leq n, q \leq 100\,000$).

В следующих n строках даны по два целых числа L_i и R_i ($0 \leq L_i, R_i \leq n$) — номер левого и правого ребёнка вершины i соответственно, либо 0 , если соответствующий ребёнок отсутствует.

Гарантируется, что L_i и R_i задают корректное бинарное дерево.

В следующих q строках даны запросы. Первое число в строке t ($t \in \{1, 2\}$) — тип запроса.

В случае запроса первого типа далее даны целые числа l, r и x ($1 \leq l \leq r \leq n$, x равен $-1, 0$ или 1) — границы отрезка вершин, в которых меняются числа, и новое значение.

В случае запроса второго типа далее дано число i ($1 \leq i \leq n$) — номер вершины, позицию которой в обходе необходимо вывести.

Формат выходных данных

На каждый запрос второго типа выведите единственное число от 1 до n — позицию соответствующей вершины в обходе.

Система оценки

Пусть q_1 — количество запросов первого типа.

Подзадача	Баллы	Дополнительные ограничения	Необх. подзадачи	Информация о проверке
1	10	$n, q \leq 5000$		первая ошибка
2	5	$q_1 \leq 10$		первая ошибка
3	10	все запросы первого типа идут до всех запросов второго типа		первая ошибка
4	10	все листья (вершины без детей) находятся на одном расстоянии от корня, нет вершин с ровно одним ребёнком		первая ошибка
5	10	$l = r$ для всех запросов первого типа		первая ошибка
6	20	$x \in \{-1, 1\}$ для всех запросов первого типа, у каждой вершины не более одного ребёнка		первая ошибка
7	10	$x \in \{-1, 1\}$ для всех запросов первого типа	6	первая ошибка
8	10	у каждой вершины не более одного ребёнка	6	первая ошибка
9	15	нет	1–8	первая ошибка

Пример

стандартный ввод	стандартный вывод
5 5	4
3 4	1
0 0	2
5 2	
0 0	
0 0	
2 2	
1 1 3 1	
2 5	
1 3 3 0	
2 3	

Замечание

В примере обход меняется следующим образом:

- [1, 3, 5, 2, 4]
- [5, 2, 3, 4, 1]
- [5, 3, 2, 4, 1]