

# Проверка сбалансированности

АВЛ-дерево является сбалансированным в следующем смысле: для любой вершины высота ее левого поддерева отличается от высоты ее правого поддерева не больше, чем на единицу.

Введем понятие баланса вершины: для вершины дерева  $V$  ее баланс  $B(V)$  равен разности высоты правого поддерева и высоты левого поддерева.

Таким образом, свойство АВЛ-дерева, приведенное выше, можно сформулировать следующим образом: для любой ее вершины  $V$  выполняется следующее неравенство:  $-1 \leq B(V) \leq 1$ .

Дано двоичное дерево поиска. Для каждой его вершины требуется определить ее баланс.

Входные данные

Входные данные содержат описание двоичного дерева.

В первой строке файла находится число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) - число вершин в дереве. В последующих  $n$  строках файла находятся описания вершин дерева. В  $(i+1)$ -ой строке файла находится описание  $i$ -ой вершины, состоящее из трех чисел  $K_i$ ,  $L_i$ ,  $R_i$ , разделенных пробелами: ключа в  $i$ -ой вершине ( $|K_i| \leq 10^9$ ), номера левого ребенка  $i$ -ой вершины ( $i < L_i \leq n$  или  $L_i = 0$ , если левого ребенка нет) и номера правого ребенка  $i$ -ой вершины ( $i < R_i \leq n$  или  $R_i = 0$ , если правого ребенка нет).

Все ключи различны. Гарантируется, что данное дерево является деревом поиска.

Выходные данные

Для  $i$ -ой вершины в  $i$ -ой строке выведите одно число - баланс данной вершины.

## Делаю я левый поворот

Для балансировки AVL-дерева при операциях вставки и удаления производятся левые и правые повороты. Левый поворот в вершине производится, когда баланс этой вершины больше 1, аналогично, правый поворот производится при балансе, меньшем 1.

Дано дерево, в котором баланс дерева равен 2. Сделайте левый поворот.

Входные данные

Входные данные содержат описание двоичного дерева.

В первой строке файла находится число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) - число вершин в дереве. В последующих  $n$  строках файла находятся описания вершин дерева. В  $(i+1)$ -ой строке файла находится описание  $i$ -ой вершины, состоящее из трех чисел  $K_i, L_i, R_i$ , разделенных пробелами: ключа в  $i$ -ой вершине ( $|K_i| \leq 10^9$ ), номера левого ребенка  $i$ -ой вершины ( $i < L_i \leq n$  или  $L_i = 0$ , если левого ребенка нет) и номера правого ребенка  $i$ -ой вершины ( $i < R_i \leq n$  или  $R_i = 0$ , если правого ребенка нет).

Все ключи различны. Гарантируется, что данное дерево является деревом поиска. Все ключи различны. Баланс корня дерева (вершины с номером 1) равен 2, баланс всех остальных вершин находится в пределах от -1 до 1.

Выходные данные

Выведите в том же формате дерево после осуществления левого поворота. Нумерация вершин может быть произвольной при условии соблюдения формата. Так, номер вершины должен быть меньше номера ее детей.

## Вставка в AVL-дерево

Вставка в AVL-дерево вершины  $V$  с ключом  $X$  при условии, что такой вершины в этом дереве нет, осуществляется следующим образом:

- находится вершина  $W$ , ребенком которой должна стать вершина  $V$ ;
- вершина  $V$  делается ребенком вершины  $W$ ;
- производится подъем от вершины  $W$  к корню, при этом, если какая-то из вершин несбалансированна, производится, в зависимости от значения баланса, левый или правый поворот.

Первый этап нуждается в пояснении. Спуск до будущего родителя вершины  $V$  осуществляется, начиная от корня, следующим образом:

Пусть ключ текущей вершины равен  $Y$ .

- Если  $X < Y$  и у текущей вершины есть левый ребенок, переходим к левому ребенку.
- Если  $X < Y$  и у текущей вершины нет левого ребенка, то останавливаемся, текущая вершина будет родителем новой вершины.
- Если  $X > Y$  и у текущей вершины есть правый ребенок, переходим к правому ребенку.
- Если  $X > Y$  и у текущей вершины нет правого ребенка, то останавливаемся, текущая вершина будет родителем новой вершины.

Отдельно рассматривается следующий крайний случай если до вставки дерево было пустым, то вставка новой вершины осуществляется проще: новая вершина становится корнем дерева.

Входные данные

Входные данные содержат описание двоичного дерева.

В первой строке файла находится число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) - число вершин в дереве. В последующих  $n$  строках файла находятся описания вершин дерева. В  $(i+1)$ -ой строке файла ( $1 \leq i \leq n$ ) находится описание  $i$ -ой вершины, состоящее из трех чисел  $K_i, L_i, R_i$ , разделенных пробелами ключа в  $i$ -ой вершине ( $|K_i| \leq 10^9$ ), номера левого ребенка  $i$ -ой вершины ( $i < L_i \leq n$  или  $L_i = 0$ , если левого ребенка нет) и номера правого ребенка  $i$ -ой вершины ( $i < R_i \leq n$  или  $R_i = 0$ , если правого ребенка нет).

Все ключи различны. Гарантируется, что данное дерево является корректным AVL-деревом.

В последней строке содержится число  $X$  ( $|X| \leq 10^9$ ) ключ вершины, которую требуется вставить в дерево. Гарантируется, что такой вершины в дереве нет.

Выходные данные

Выведите в том же формате дерево после осуществления левого поворота. Нумерация вершин может быть произвольной при условии соблюдения формата. Так, номер вершины должен быть меньше номера ее детей.

## Три друга 2: возвращение

Три друга списывают лабораторную работу, каждый из них списывает по  $n$  различных задач. Поскольку друзья не очень умные, они не меняют названия отправляемых на проверку файлов.

По истечении времени, отведенного на написание лабораторной, преподаватель запускает бан-машину и ставит баллы по следующим правилам:

- если задача написана только у одного студента, то этот студент получает 3 балла, поскольку эту задачу он не списывал и не давал списывать;
- если задача списана ровно у двух студентов, то каждый из них получает по 1 утешительному баллу;
- если задача списана всеми тремя студентами, то за нее баллы не начисляются никому.

Выведите финальное количество баллов у каждого студента.

В рамках этой задачи будем считать, что Бан-машина считает решения списанными, если у них полностью совпадают имена файлов.

Входные данные

В первой строке входных данных дается число  $n$  ( $1 \leq n \leq 100000$ ) - количество задач в лабораторной.

Следующие три строки содержат по  $n$  различных слов в каждой — названия файлов с решениями, отправленных каждым из студентов.

Выходные данные

Необходимо вывести 3 числа - количество баллов у первого, второго и третьего студента соответственно.

## Упорядоченное множество на AVL-дереве

Если Вы сдали все предыдущие задачи, Вы уже можете написать эффективную реализацию упорядоченного множества на AVL-дереве. Сделайте это.

Для проверки того, что множество реализовано именно на AVL-дереве, мы просим Вас выводить баланс корня после каждой операции вставки и удаления.

Операции вставки и удаления требуется реализовать точно так же, как это было сделано в предыдущих двух задачах, потому что в ином случае баланс корня может отличаться от требуемого.

Входные данные

В первой строке находится число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) - число операций над множеством. Изначально множество пусто. В каждой из последующих  $n$  строк находится описание операции. Операции бывают следующих видов:

- $A\ x$  - вставить число  $x$  в множество. Если число  $x$  там уже содержится, множество изменять не следует.
- $D\ x$  - удалить число  $x$  из множества. Если числа  $x$  нет в множестве, множество изменять не следует.

- $s\ x$  - проверить, есть ли число  $x$  в множестве.

Выходные данные

Для каждой операции вида  $s\ x$  выведите  $y$ , если число  $x$  содержится в множестве, и  $n$ , если не содержится. Для каждой операции вида  $a\ x$  или  $d\ x$  выведите баланс корня дерева после выполнения операции. Если дерево пустое (в нем нет вершин), выведите 0.

Вывод для каждой операции должен содержаться на отдельной строке.

## Очистка телефона

У Пети в телефоне записаны номера друзей. Однажды Петя заметил, что у него кончается свободная память на устройстве. Петя знал, что среди его контактов есть множество дубликатов. Он решил удалить повторяющиеся записи.

Каждый контакт - это строка, состоящая исключительно из строчных или заглавных букв латинского алфавита и цифр. При удалении строки освобождается  $M$  байт памяти, где  $M$  - количество различных символов в данной строке. Помогите Пете узнать, сколько памяти в байтах он сможет освободить.

Входные данные

В первой строке входных данных дается число  $n$  ( $1 \leq n \leq 10^5$ ) - количество контактов в телефоне.

В следующей строке записаны через пробел  $n$  строк - контакты Пети.

Выходные данные

Выведите одно число - ответ на задачу.

## Петя кормит котов

Мама подарила мальчику Пете  $n$  кошек.

Что бы кошки могли спокойно кушать, Петя завел  $n$  мисок для корма. Миска с номером  $j$  принадлежит кошке с номером  $j$ . Изначально миски пустые.

Существует 3 вида событий:

1. Кошка с номером  $i$  решила пообедать и съела из своей миски  $k$  кусочков мяса. Если в миске не хватает мяса, то кошка просто съедает все, что там есть и остается голодной.
2. Петя может в миску с номером  $i$  положить  $k$  кусочков мяса.
3. Мама, для учета расходов, решила спросить сколько кусочков мяса в сумме лежат в мисках с номерами от  $L$  до  $R$ .

Входные данные

В первой строке дается  $1 \leq N \leq 5 \cdot 10^5$  и  $1 \leq Q \leq 10^5$  - количество кошек и событий. В следующих  $Q$  строчках идет описание событий:

1)+  $i$   $k$  добавить в миску под номером  $i$   $k$  кусочков мяса.

2)-  $i$   $k$  кошка под номером  $i$  скушала  $k$  кусочков мяса из своей миски.

3)?  $L$   $R$  мама спросила у Пети сколько кусочков мяса в сумме лежит в мисках от  $L$  до  $R$  (включительно).

$1 \leq i, L, R \leq n; 1 \leq k \leq 1000$ .

Выходные данные

На каждый запрос мамы выведите в новой строке одно число - сумму кусочков мяса в интересующих её мисках.