

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 ($1 \leq i \leq n$) 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 деревом поиска.

Выходные данные

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

STDIN

```
6
-2 0 2
8 4 3
9 0 0
3 5 6
0 0 0
6 0 0
```

STDOUT

```
3
-1
0
0
0
0
```

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.

Существует два разных левых (как, разумеется, и правых) поворота: большой и малый левый поворот.

Малый левый поворот осуществляется следующим образом:

{скоро тут будет иллюстрация}

Заметим, что если до выполнения малого левого поворота был нарушен баланс только корня дерева, то после его выполнения все вершины становятся сбалансированными, за исключением случая, когда у правого ребенка корня баланс до поворота равен 1. В этом случае вместо малого левого поворота выполняется большой левый поворот, который осуществляется так:

{скоро тут будет иллюстрация}

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 аила ($1 \leq i \leq n$) 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 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.

STDIN

```
7
-2 7 2
8 4 3
9 0 0
3 5 6
0 0 0
6 0 0
-7 0 0
```

STDOUT

```
7
3 2 3
-2 4 5
8 6 7
-7 0 0
0 0 0
6 0 0
9 0 0
```

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 несбалансирована, 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 аила ($1 \leq i \leq n$) 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 АВЛ-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.

STDIN

```
2
3 0 2
4 0 0
5
```

STDOUT

```
3
4 2 3
3 0 0
5 0 0
```

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.

STDIN

```
3
fir sec thi
thi fir sec
aaa sec bbb
```

STDOUT

```
2 2 6
```