Содержание

Задача 7A. Двоичное дерево поиска [2 секунды, 256 мегабайт]

Реализуйте сбалансированное двоичное дерево поиска.

Формат входных данных

Входной файл содержит описание операций с деревом. Операций не больше 10^5.
В каждой строке находится одна из следующих операций:

- `insert` x — добавить в дерево ключ x.
- `delete` x — удалить из дерева ключ x.
  Если ключа x в дереве нет, то ничего делать не надо.
- `exists` x — если ключ x есть в дереве, «`true`», иначе «`false`»
- `next` x — минимальный элемент в дереве, $> x$, или «`none`», если такого нет.
- `prev` x — максимальный элемент в дереве, $< x$, или «`none`», если такого нет.

Формат выходных данных

Выведите последовательно результат выполнения всех операций `exists`, `next`, `prev`.
Следуйте формату выходного файла из примера.

Примеры

bst.in	bst.out
insert 2 insert 5 insert 3 exists 2 exists 4 next 4 prev 4 delete 5 next 4 prev 4	true false 5 3 none 3

Замечание

Don't pause just Splay.

Задача 7B. Динамический Лес [2 секунды, 256 мебибайт]

Вам нужно научиться обрабатывать 3 типа запросов:

1. Добавить ребро в граф (`link`).
2. Удалить ребро из графа (`cut`).
3. По двум вершинам a и b вернуть длину пути между ними (или -1, если они лежат в разных компонентах связности) (`get`).

Изначально граф пустой (содержит N вершин, не содержит ребер). Гарантируется, что в любой момент времени граф является лесом. При добавлении ребра гарантируется, что его сейчас в графе нет. При удалении ребра гарантируется, что оно уже добавлено.

Формат входных данных

Числа N и M ($1 \leqslant N \leqslant 10^5 + 1$, $1 \leqslant M \leqslant 10^5$) — количество вершин в дереве и, соответственно, запросов. Далее M строк, в каждой строке команда (`link` или `cut`, или `get`) и 2 числа от 1 до N — номера вершин в запросе.

Формат выходных данных

В выходной файл для каждого запроса `get` выведите одно число — расстояние между вершинами, или -1, если они лежат в разных компонентах связности.

Пример

стандартный ввод	стандартный вывод
3 7 get 1 2 link 1 2 get 1 2 cut 1 2 get 1 2 link 1 2 get 1 2	-1 1 -1 1
5 10 link 1 2 link 2 3 link 4 3 cut 3 4 get 1 2 get 1 3 get 1 4 get 2 3 get 2 4 get 3 4	1 2 -1 1 -1 -1

Замечание

Можно Splay, можно просто декартово дерево. Существуют решения и без link/cut, но давайте нет.

Задача 7C. Волшебный лес [2 секунды, 512 мегабайт]

На планете Black Lizard есть волшебный лес, представляющий из себя неориентированный граф из n вершин и m рёбер.

Анджи хочет пройти из вершины 1 в n. У Анджи есть уровень силы A и энергии B ($A, B \geqslant 0$).

Каждое ребро графа характеризуется концами v_i, u_i и числами a_i и b_i. Анджи может пройти по ребру только если $A \geqslant a_i$ и $B \geqslant b_i$.

Найдите A и B с минимальной суммой, позволяющие пройти Анджи через лес.

Формат входных данных

Первая строка содержит n и m ($2 \leqslant n \leqslant 50\,000$, $0 \leqslant m \leqslant 100\,000$).

Следующие m строк описывают рёбра и состоят из четырёх целых чисел каждая v_i, u_i, a_i, b_i ($1 \leqslant v_i, u_i \leqslant n$, $1 \leqslant a_i, b_i \leqslant 50\,000$).

Граф может содержать кратные рёбра и петли.

Формат выходных данных

Выведите минимальную сумму $A + B$. Если пройти никак не получается, то выведите «-1».

Пример

стандартный ввод	стандартный вывод
4 5 1 2 19 1 2 3 8 12 2 4 12 15 1 3 17 8 3 4 1 17	32
3 1 1 2 1 1	-1