Задача про косое дерево

Реализуйте косое дерево (splay tree).

Реализация самой структуры данных должна быть инкапуслирована, т.е. не зависеть от форматов входных/выходных данных и непосредственно ввода/вывода.

Тесты предполагают "левостороннюю" реализацию, т.е. если действие можно реализовать двумя симметричными способами, надо делать тот, который больше использует левую сторону.

Формат ввода

На стандартном потоке ввода задаётся последовательность команд. Пустые строки игнорируются.

Каждая строка содержит ровно одну команду: add K V, set K V, delete K, search K, min, max или print, где K - целое число (64 бита вам хватит), ключ, V - произвольная строка без пробелов (значение).

Формат вывода

Команда add добавляет значение V в дерево по ключу K, set - изменяет данные по ключу, команда delete удаляет данные.

Команда search выводит либо "1 V", либо "0", где V - значение для найденного ключа.

Команды min и max выводят "K V", где K - минимальный или максимальный ключ дерева соответственно, V - значение по этому ключу.

Команда print выводит все дерево целиком. Она не изменяет дерево.

Дерево выводится строго по уровням, слева направо, 1 строка - 1 уровень. Первая строка содержит только корень дерева в формате "[K V]" или "_", если дерево пустое.

Каждая последующая строка содержит один уровень дерева. Вершины выводятся в формате "[K V P]", где P - ключ родительской вершины. Если вершина отсутствует, ставится "_". Вершины разделены пробелом.

В любой непонятной ситуации результатом работы любой команды будет "error".

Результат работы программы выводится в стандартный поток вывода.

Пример

Ввод:

```
add 8 10
```

```
add 4 14
add 7 15
set 8 11
add 3 13
add 5 16
search 88
search 7
delete 5
print
```

Вывод:

```
0
1 15
[4 14]
[3 13 4] [7 15 4]
_ _ _ [8 11 7]
```