Непростая куча

Реализуйте двоичную min-кучу. Модифицируйте ее таким образом, чтобы внутреннее ее строение было таким же, но при этом доступ по ключу к любому элементу осуществлялся в среднем за константное время.

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

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, extract 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 кучу по ключу K, set - изменяет данные по ключу, команда delete удаляет данные.

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

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

Команда extract извлекает корень кучи и выводит "K V", где K, V - ключ и значение извлеченного элемента.

Команда print выводит всю кучу целиком.

Куча выводится строго по уровням, слева направо, 1 строка - 1 уровень. Первая строка содержит только корень кучи в формате "[K V]" или "_", если куча пустая.

Каждая последующая строка содержит один уровень кучи. 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
add 10 10
search 88
search 7
delete 4
extract
print
```

Вывод:

```
0
1 2 15
3 13
[5 16]
[8 11 5] [7 15 5]
[10 10 8] _ _ _
```