

Для решения этой задачи можно использовать декартово дерево. Вставка и удаление будут реализованы так же, как и в декартовом дереве с явным ключом.

А вместо последней операции реализуем такую, которая просто находит минимальный элемент, которого нет в множестве. Для того чтобы эту операцию свести к исходной, нужно применить операцию *split* (причем реализация этой функции должна быть, такая, в которой вершины со ключом x находятся в правом поддереве) к исходному дереву, деля по ключу, который является аргументом. Тогда если применить нашу функцию нахождения минимального элемента, которого нет в множестве, к правому поддереву, то получим нужный результат — минимальный элемент, меньший либо равный x , которого нет в множестве. Реализация этой функции будет такова:

1. Реализация будет рекурсивной, в аргументах будет вершина дерева, которую сейчас обрабатываем и указатель на минимальный элемент, которого нет в множестве.
2. Изначально будем считать, что ответ — x , а первой обрабатываемой вершиной будет корень поддерева, где все ключи будут больше x .
3. Если мы пытаемся перейти в несуществующую вершину, то значит, что найден ответ, вернем указатель на ответ.
4. Иначе:
 - (a) Если в левом поддереве не содержится все ключи, которые принадлежат отрезку $[x; \text{ключ обрабатываемой вершины}]$ (Его размер не равен ключ - указатель на минимальный ответ), то перейдем в левое поддерево, не передвигая указатель.
 - (b) Иначе, перейдем в правое поддерево, переместив указатель минимального возможного ответа на ключ обрабатываемой вершины $+1$.