1 $2-d$ деревья

Вершины этих деревьев имеют следующий тип.
$nodetype = record$
$INFO : infotype$
$XVAL : real$
$YVAL : real$
$LLINK :\uparrow nodetype$
$RLINK :\uparrow nodetype$
end
Здесь поле INFO содержит произвольную пользовательскую информацию (например, географическое название места), поля XVAL и YVAL - координаты точки, связанной с вершиной, LLINK и RLINK - ссылки на левого и правого сыновей вершины.

Если v - вершина дерева T, то через $level(v)$ обозначим ее глубину (уровень вершины). В частности, для корня r этого дерева $level(r) = 0$.

Определение 1. 2-d деревом *называется любое бинарное дерево с вершинами типа nodetype, удовлетворяющее следующим условиям.*
1. Если v - вершина дерева, находящаяся на четном уровне ($level(v)$ - четное число), то для всякой вершины w, находящейся в поддереве с корнем $v.LLINK$, $w.XVAL < v.XVAL$, а для всякой вершины w, находящейся в поддереве с корнем $v.RLINK$, $w.XVAL \geq v.XVAL$.
2. Если v - вершина дерева, находящаяся на нечетном уровне, то для всякой вершины w, находящейся в поддереве с корнем $v.LLINK$, $w.YVAL < v.YVAL$, а для всякой вершины w, находящейся в поддереве с корнем $v.RLINK$, $w.YVAL \geq v.YVAL$.

Алгоритм вставки вершины в 2-d дерево

Вход: r - корень дерева, v - вставляемая вершина.
Выход: 0 - вершина с теми же координатами имееется в дереве, 1 - нет.

INSERT2d(r,v)

begin
1. $cur := r$;
2. **if** $cur.XVAL = v.XLAL$ **and** $cur.YVAL = v.YLAL$
   then $cur.INFO := v.INFO$; $return(0)$
   end if;
3. **if** $level(cur)$ четно
4. **then if** $v.XVAL < cur.XVAL)$
5. **then if** $cur.LLINK \neq NIL$

```
6.              then  INSERT2d(cur.LLINK, v)
7.              else  cur.LLINK := v; return(1)
                end if
8.        else /* cur.XVAL ≤ v.XVAL
9.        if  cur.RLINK ≠ NIL
10.             then  INSERT2d(cur.RLINK, v)
11.             else  cur.RLINK := v; return(1)
                end if
          end if
12. else /* level(cur) нечетно
13. then if v.YVAL < cur.YVAL)
14.       then if  cur.LLINK ≠ NIL
15.             then  INSERT2d(cur.LLINK, v)
16.             else  cur.LLINK := v; return(1)
                end if
17.       else /* cur.YVAL ≤ v.YVAL
18.        if  cur.RLINK ≠ NIL
19.             then  INSERT2d(cur.RLINK, v)
20.             else  cur.RLINK := v; return(1)
                end if
          end if
    end if
    end.
```

Теорема 1.1. *Алгоритм INSERT2d(r,v) вставляет новую вершину v в 2-d дерево T с корнем r за время O(высота(T)).*

Задача 1. *Напишите алгоритм для процедуры поиска $SEARCH2d(r,(x,y))$ в 2-d дереве с корнем r, возвращающей ссылку на вершину с координатами (x,y).*

Алгоритм $DELETE2d(r,(x,y))$ удаления вершины из 2-d дерева

1) $v := SEARCH2d(r,(x,y))$;
Если v - лист, то удалить v.
Иначе пусть $level(v)$ четно.
2) Если $v.RLINK \neq NIL$, то найти в поддереве с корнем $v.RLINK$ вершину w с минимальным значением поля $XVAL$ и заменить v на w.
3) Если правого поддерева у v нет, то то найти в поддереве с корнем $v.LLINK$ вершину w с минимальным значением поля $XVAL$ и заменить v на w,
а затем "перевесить"поддерево слева направо, т.е. $v.RLINK := v.LLINK; v.LLINK := NIL$.

4) Применить (рекурсивно) $DELETE2d(w, (w.XVAL, w.YVAL))$.

Задача 2. *(а) Как изменить пункты 2 и 3 для нечетного значения* $level(v)$.
(б) Докажите, что алгоритм $DELETE2d$ *корректен.*

Задача 3. *(а) Оцените сложность алгоритма* $DELETE2d$.
(б) Измените структуру вершины, добавив поле min *со значением* (x, y) *так, что для* v *на четном уровне* $v.min$ *задает точку поддерева* T_v *с наименьшей* x*-координатой, а для* v *на нечетном уровне* $v.min$ *задает точку поддерева* T_v *с наименьшей* y*-координатой. Внесите изменения в алгоритм вставки, чтобы в нем заполнялось и поле* min.
(в) Покажите, что в структуре, описанной в п. (б) удаление можно производить за время $O(\text{высота}(T))$.

Запрос области для 2d-дерева T: по входу $(x_0, y_0), R$ выдать множество точек $A_T((x_0, y_0), R) = \{(x, y) \mid [(x, y) \in T] \ \&\ [(x_0 - x)^2 + (y_0 - y)^2 \leq R^2]\}$, т.е. множество точек из T, находящихся на расстоянии не дальше R от центра (x_0, y_0). Для эффективного ответа на такие запросы рассматривают расширение структуры вершин T за счет добавления 4-х полей: XLB, XUB, YUB, YLB таких, что для каждая точка $(x, y) \in T_v$ лежит в прямоугольнике, заданном этими координатами, т.е. $v.XLB \leq x \leq v.XUB$ и $v.YLB \leq y \leq v.YUB$ (в качестве границ допускаются $-\infty$ и $+\infty$).

Задача 4. *(1) Как изменить алгоритм вставки вершины, чтобы он определял значения указанных 4-х полей?*
(2) Предложите алгоритм ответа на запросы области, использующий расширенную структуру вершин.