Материалы презентации предназначены для размещения только для использования студентами кафедры «Компьютерные системы и технологии» НИЯУ МИФИ дневного и вечернего отделений, изучающими курс «Программирование (Алгоритмы и структуры данных)».

6. AVL – деревья

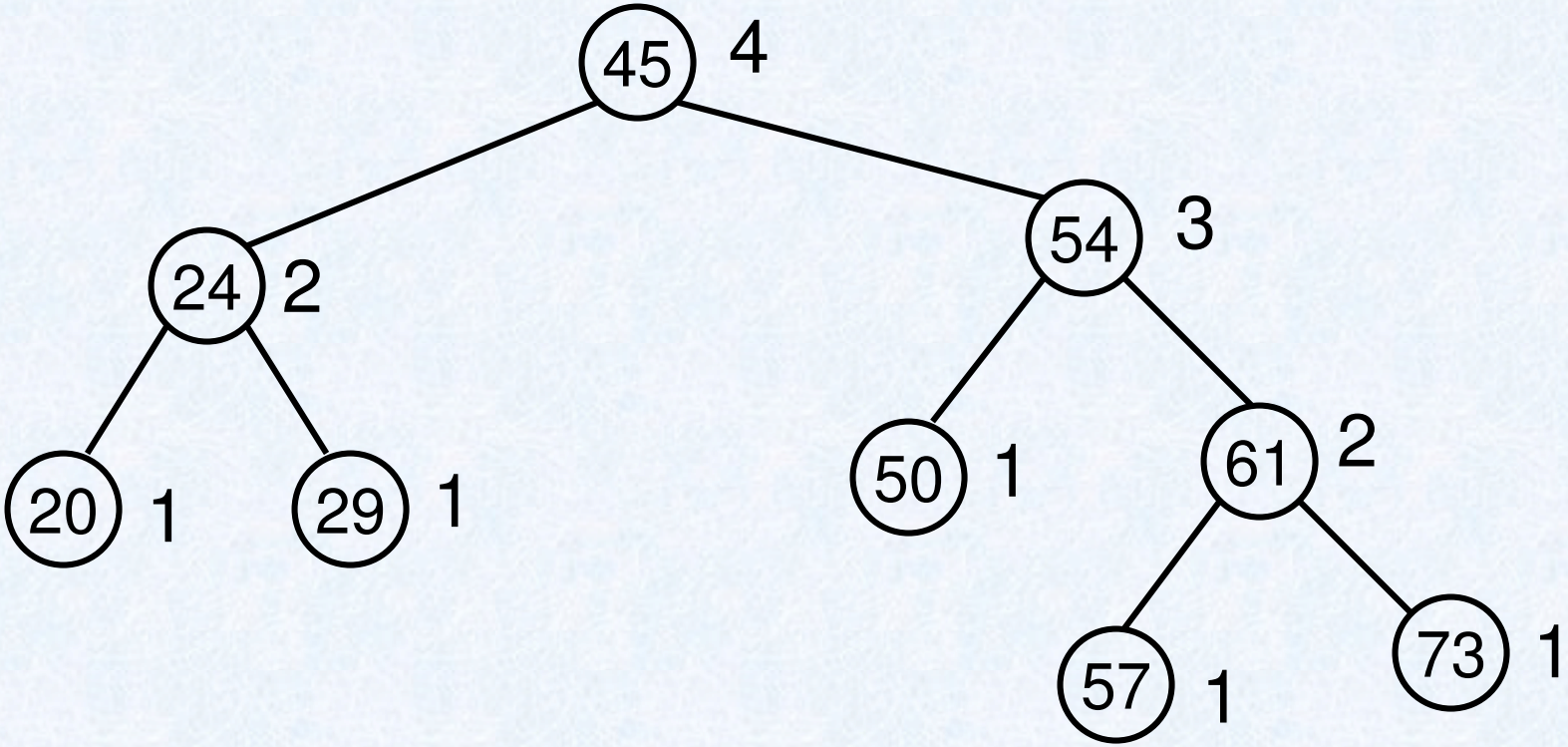
AVL – дерево

Бинарное дерево является сбалансированным, если для любого узла дерева высоты его левого и правого поддеревьев отличаются не более чем на 1

Высота дерева – максимальная длина пути от корня дерева до подчиненных листьев этого дерева

Представление листьев – как в красно-черном дереве

AVL – дерево



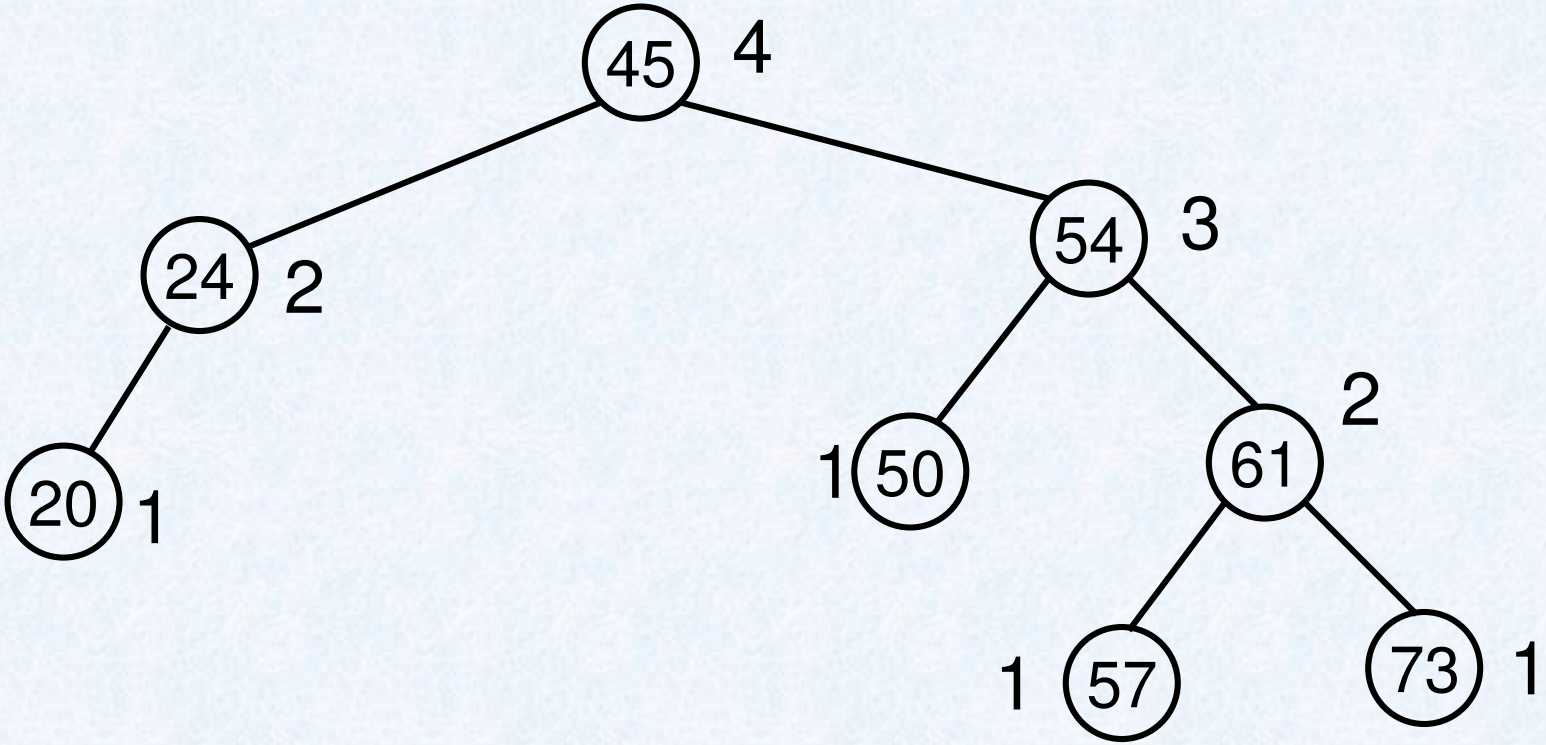
AVL – дерево

Г.М.Адельсон-Вельский, Е.М.Ландис (1962 г.)

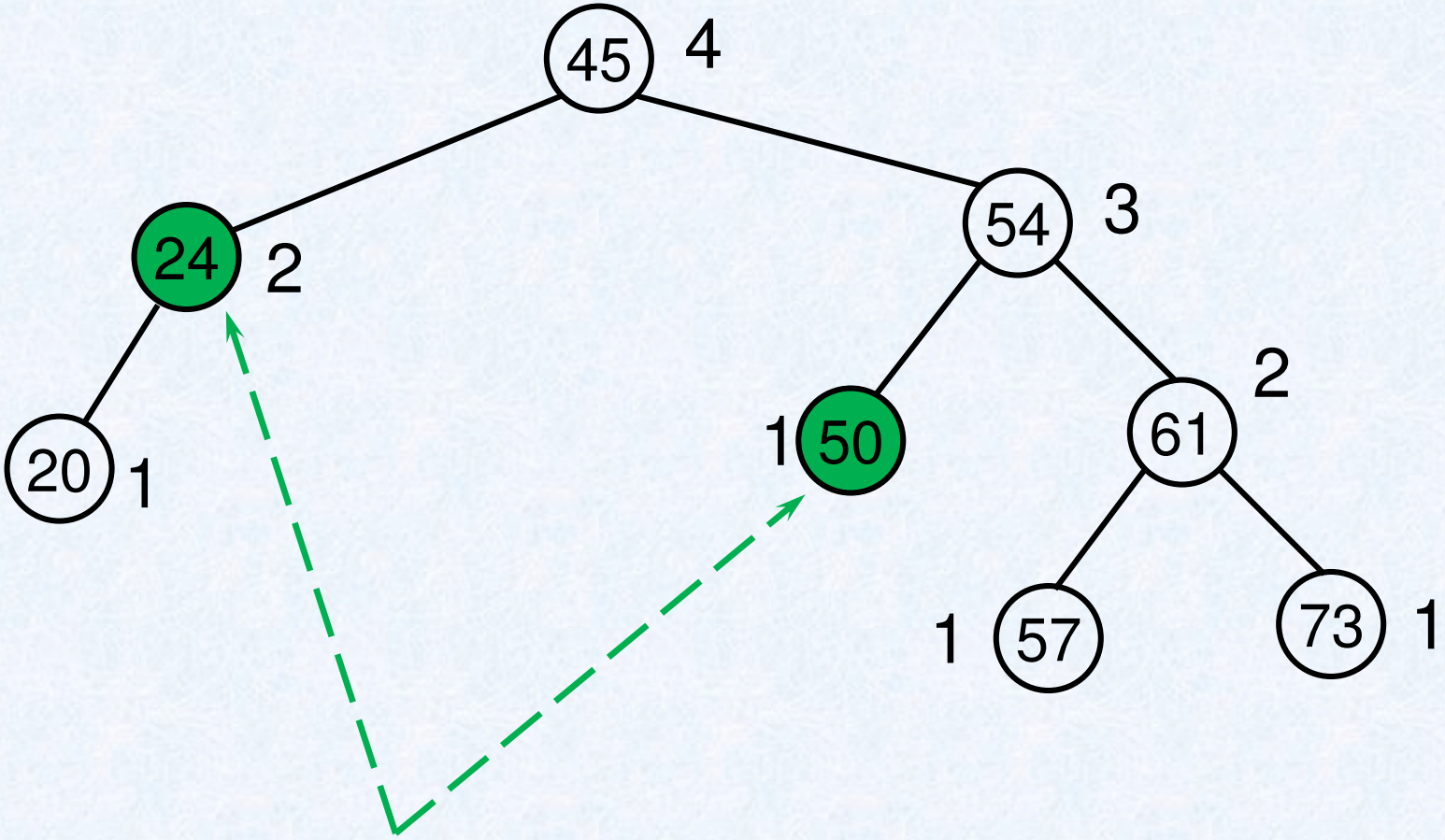
Высота сбалансированного дерева с n внутренними узлами

$$log_2(n + 1) \leq h \leq 1{,}404\ log_2(n + 2) - 0{,}328$$

Проблемы вставки

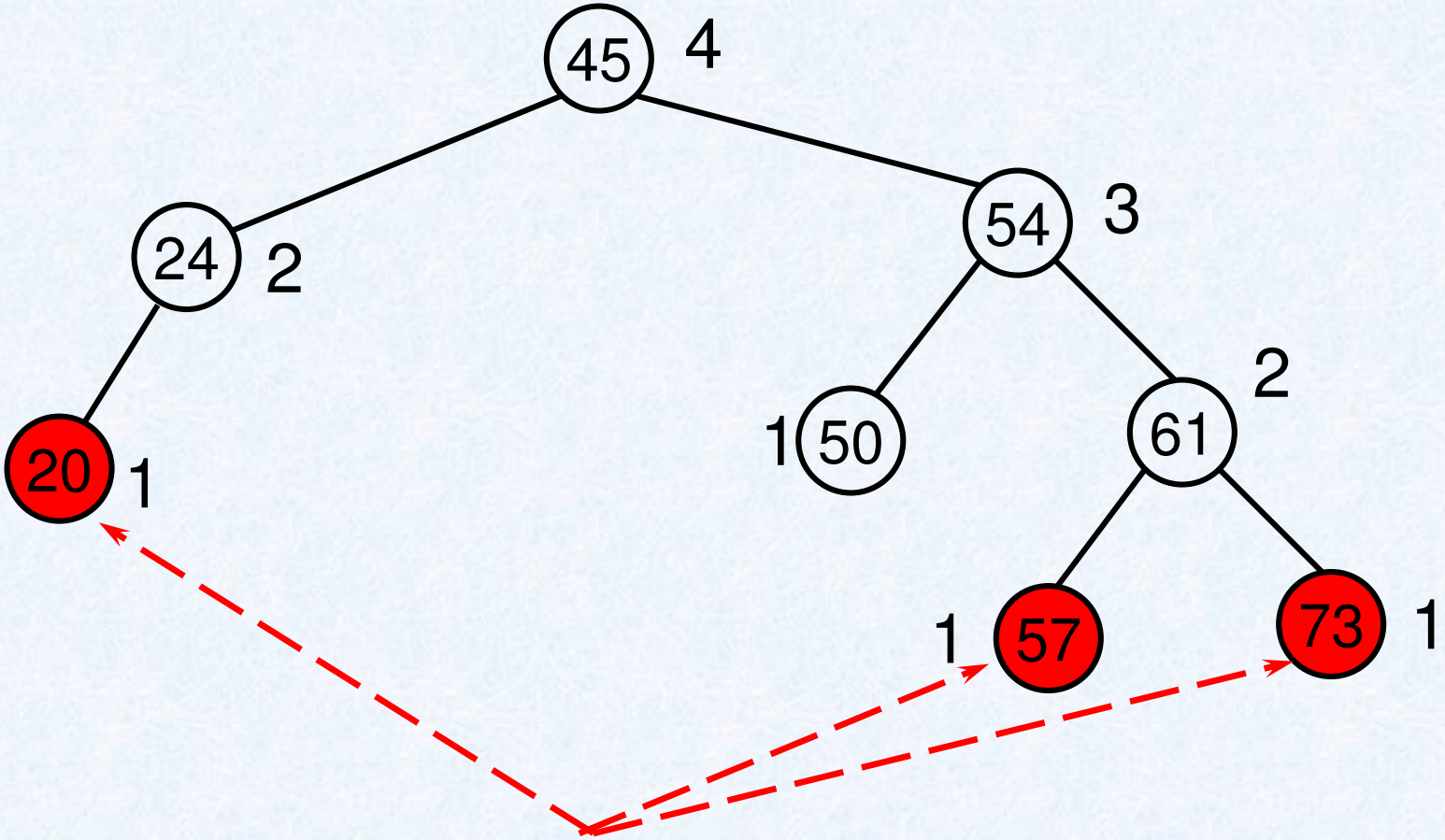


Проблемы вставки



Вставка не нарушает свойств AVL- дерева

Проблемы вставки



Вставка нарушает свойства AVL- дерева

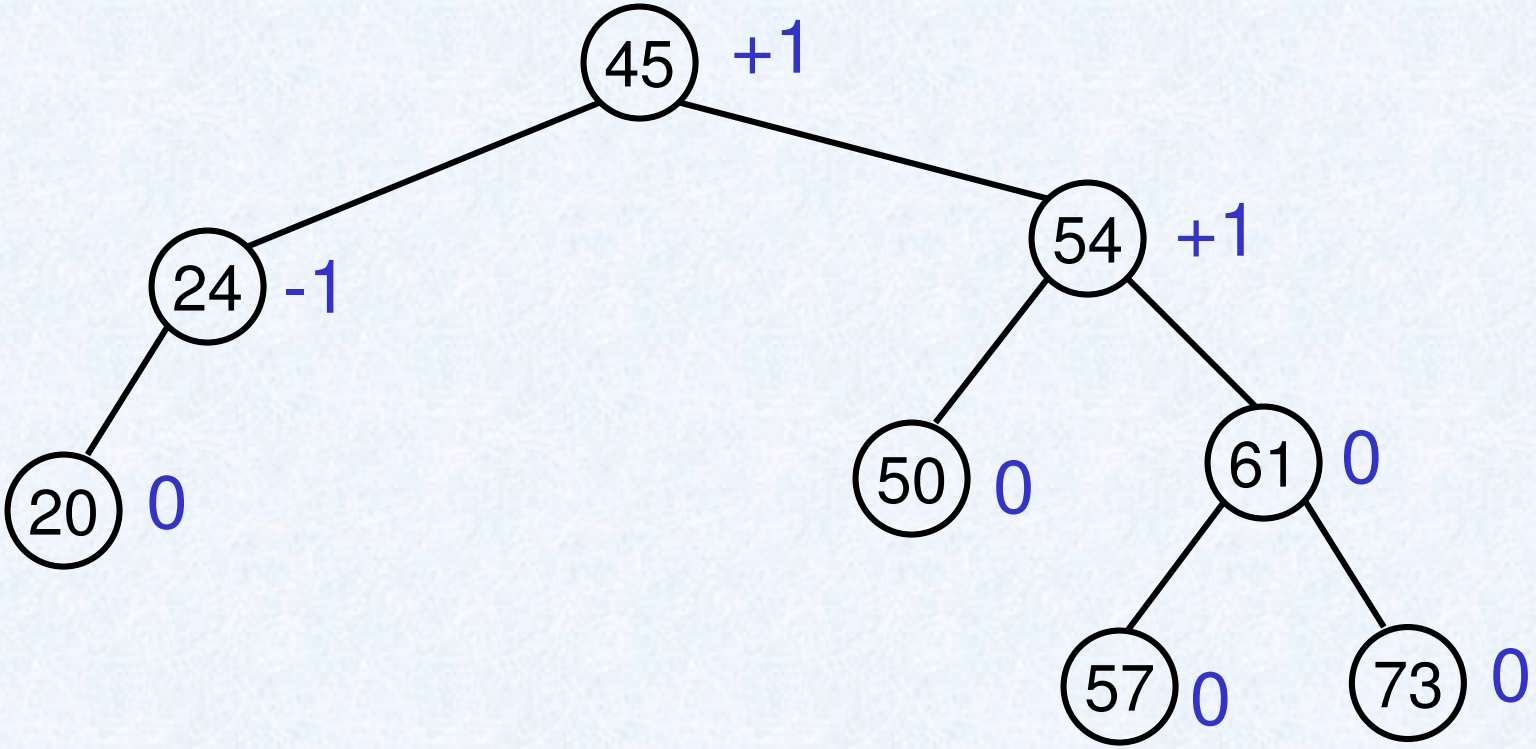
Узлы AVL- дерева

1. С дополнительным полем сбалансированности в узле:

$$balance = \begin{cases} -1, \text{ левое поддерево выше на 1} \\ 0, \text{ высота поддеревьев одинаковая} \\ +1, \text{ правое поддерево выше на 1} \end{cases}$$

```
struct Node {
   int balance;
   int key;
   struct Node *left, *right, *parent;
};
```

Узлы AVL- дерева

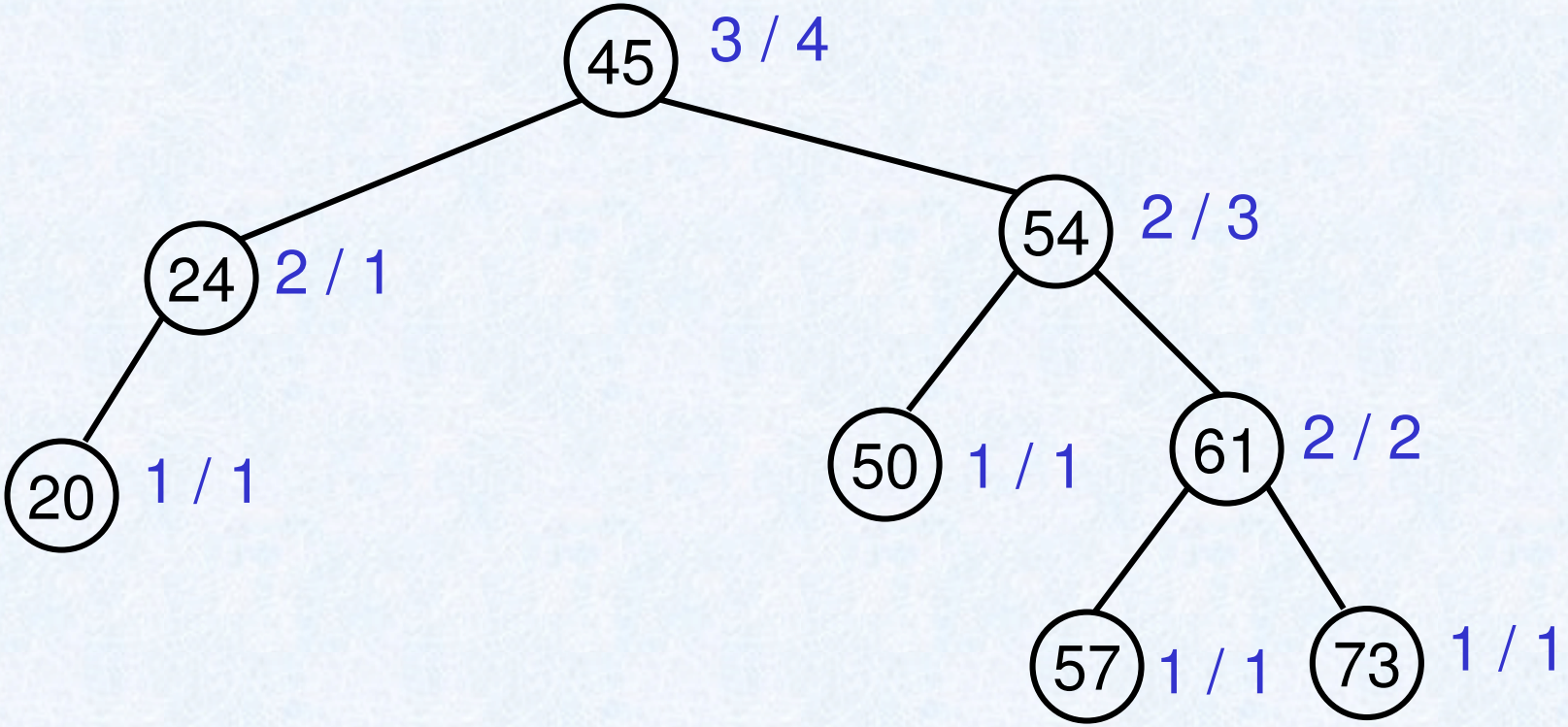


Узлы AVL- дерева

2. С дополнительными полями высоты левого и правого поддеревьев в узле:

```
struct Node {
   int hleft; – высота левого поддерева
   int hright; – высота правого поддерева
   int key;
   struct Node *left, *right, *parent;
};
```

Узлы AVL- дерева



Вставка в AVL- дерево

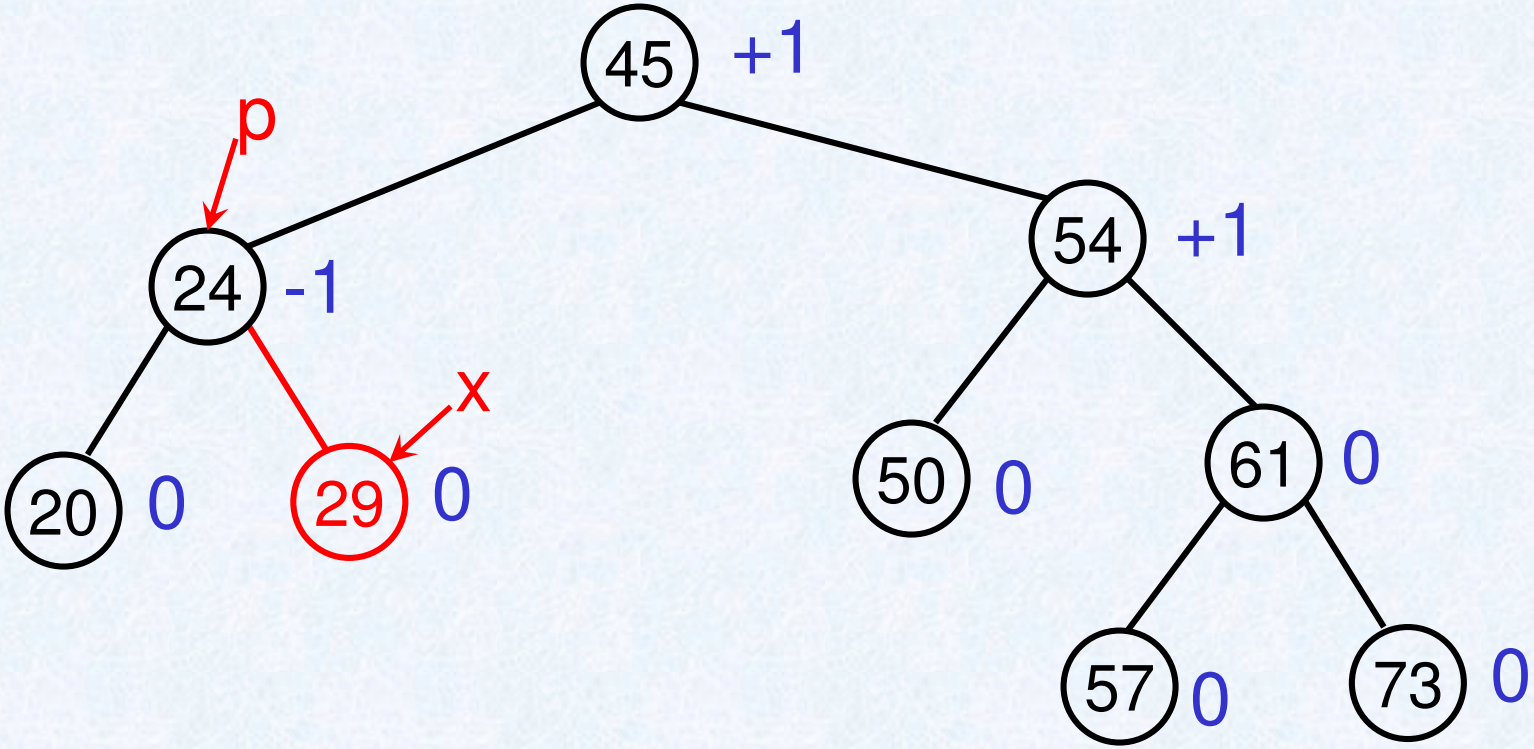
- Вставить новый элемент как в двоичное дерево поиска
- Скорректировать свойство сбалансированности узлов дерева
- При необходимости скорректировать само AVL-дерево

Коррекция узлов

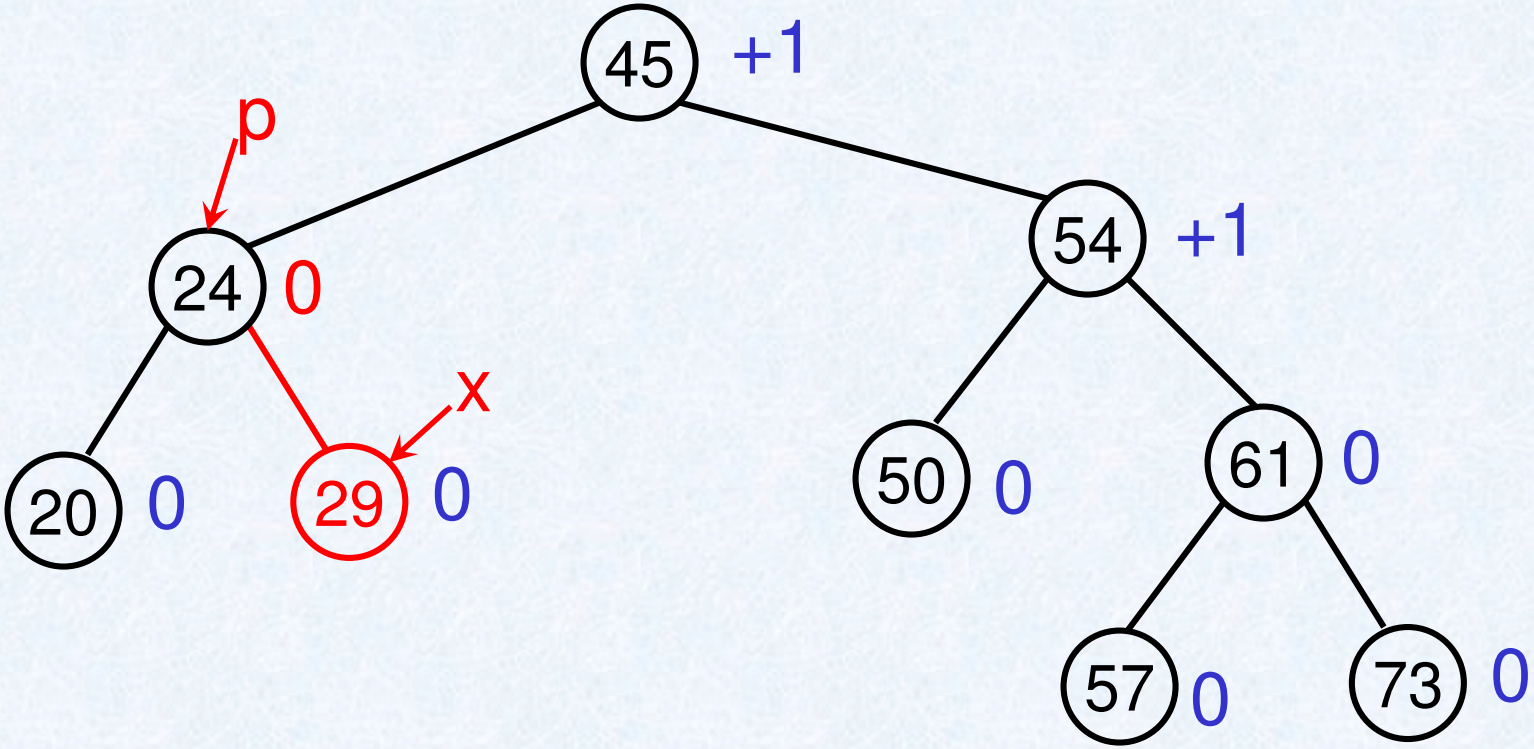
Структура узла AVL-дерева

```
struct Node {
   int balance;
   int key;
   struct Node *left, *right, *parent;
};
```

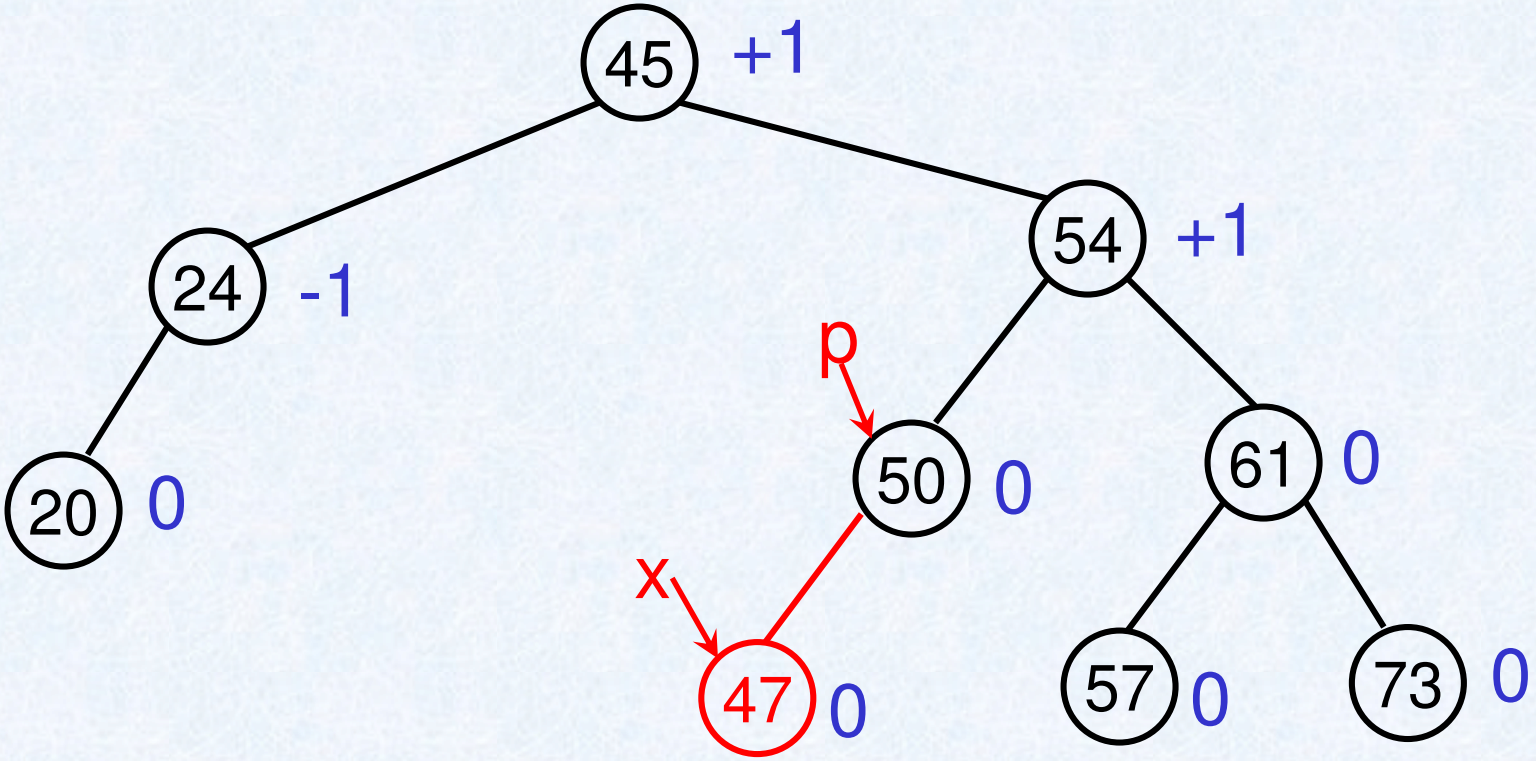
Коррекция узлов



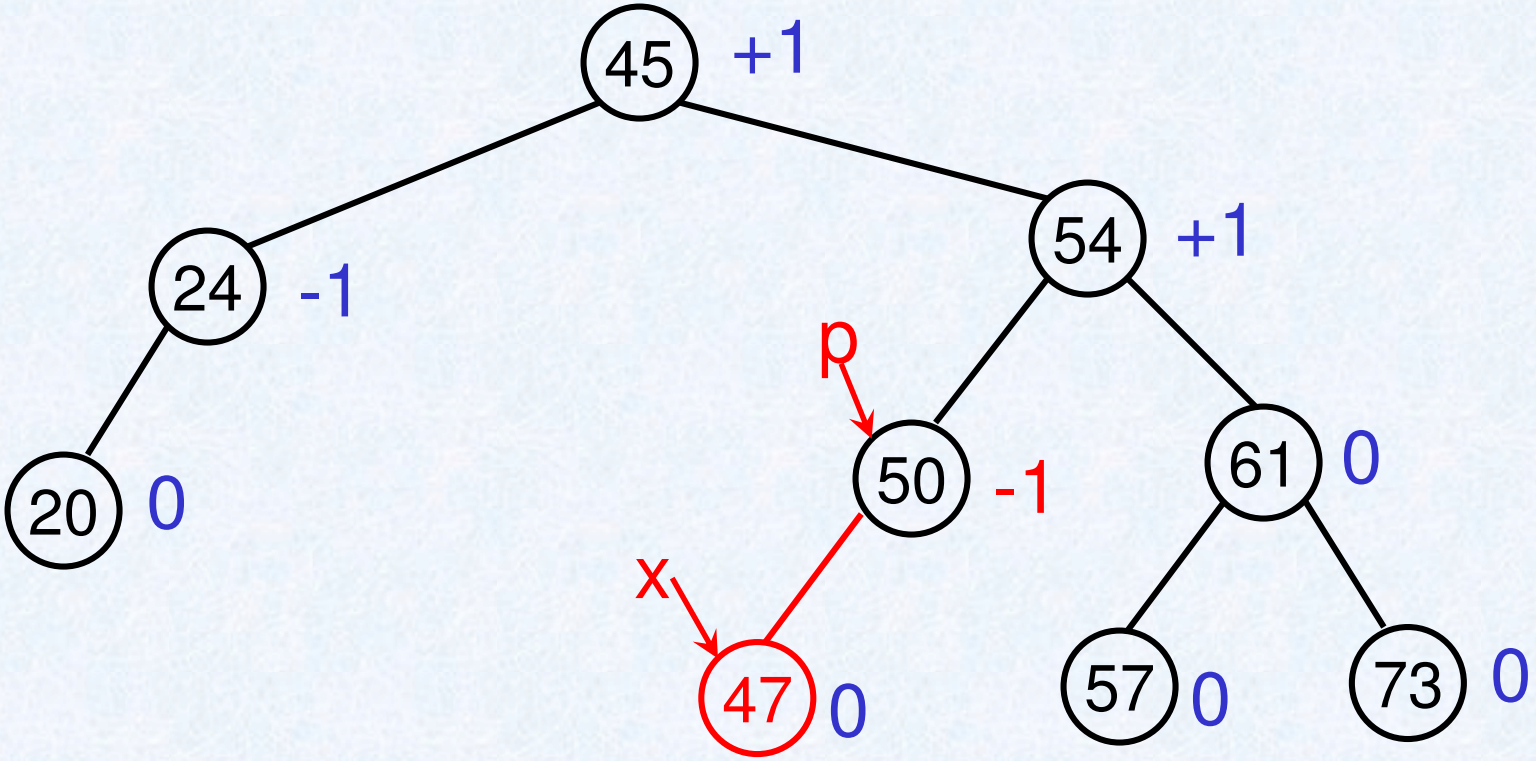
Коррекция узлов



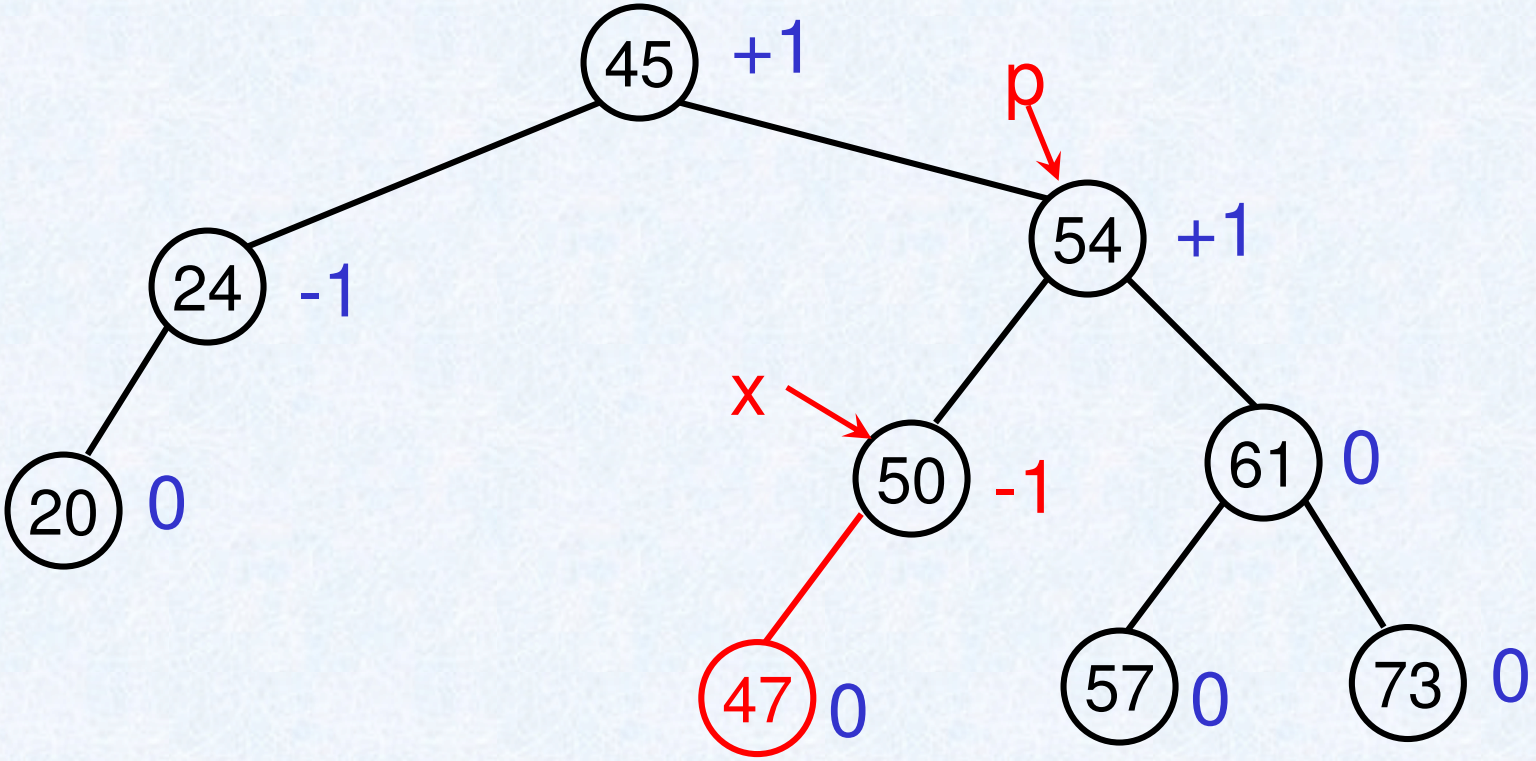
Коррекция узлов



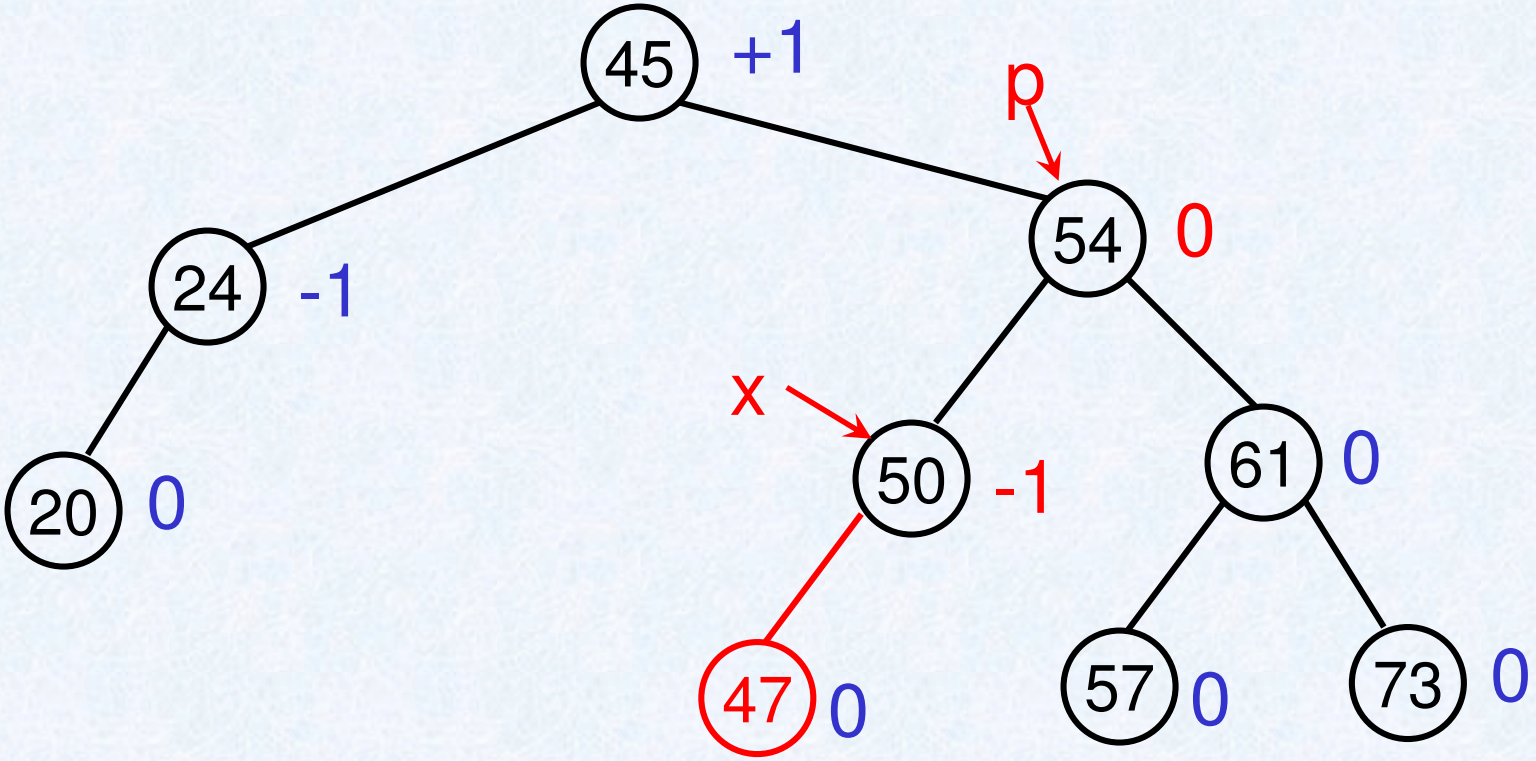
Коррекция узлов



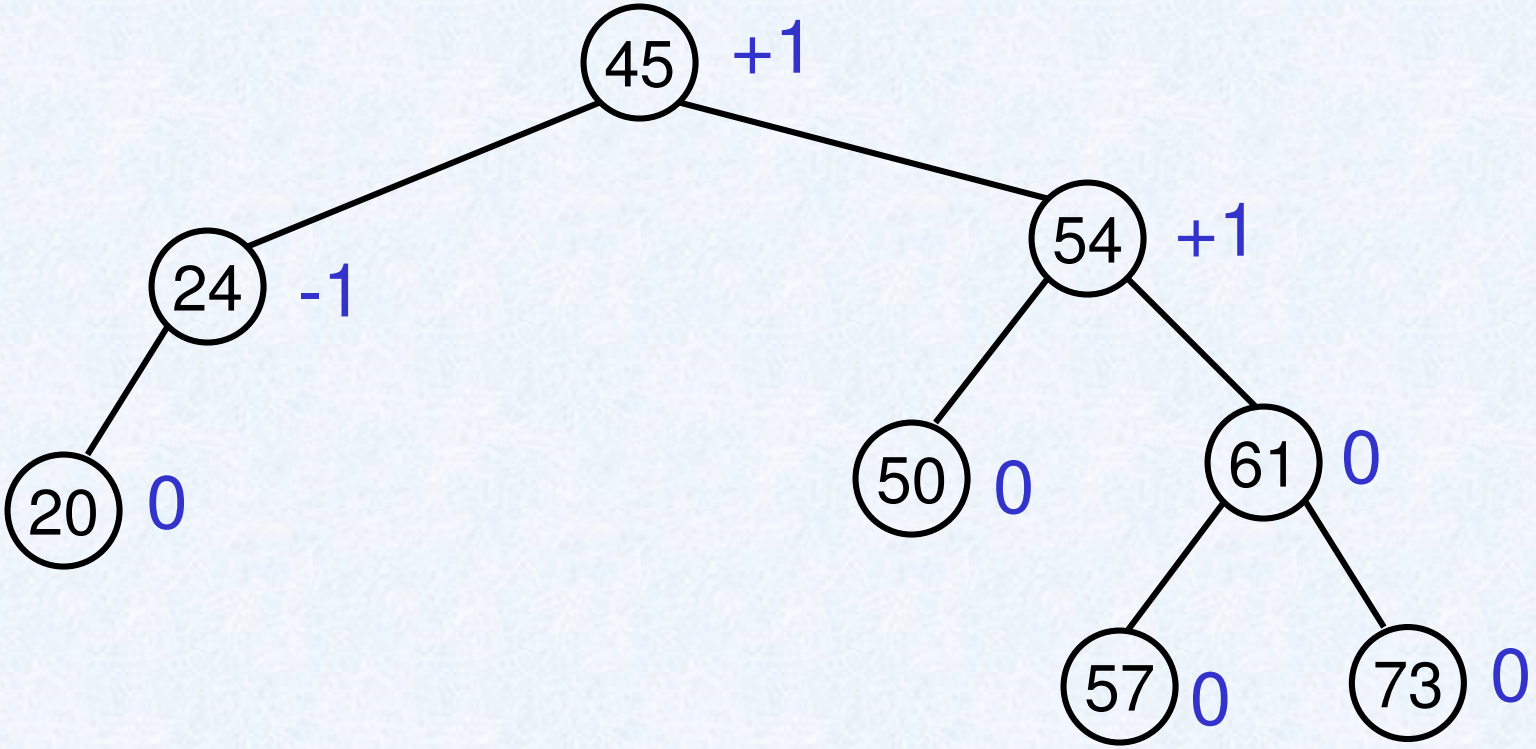
Коррекция узлов



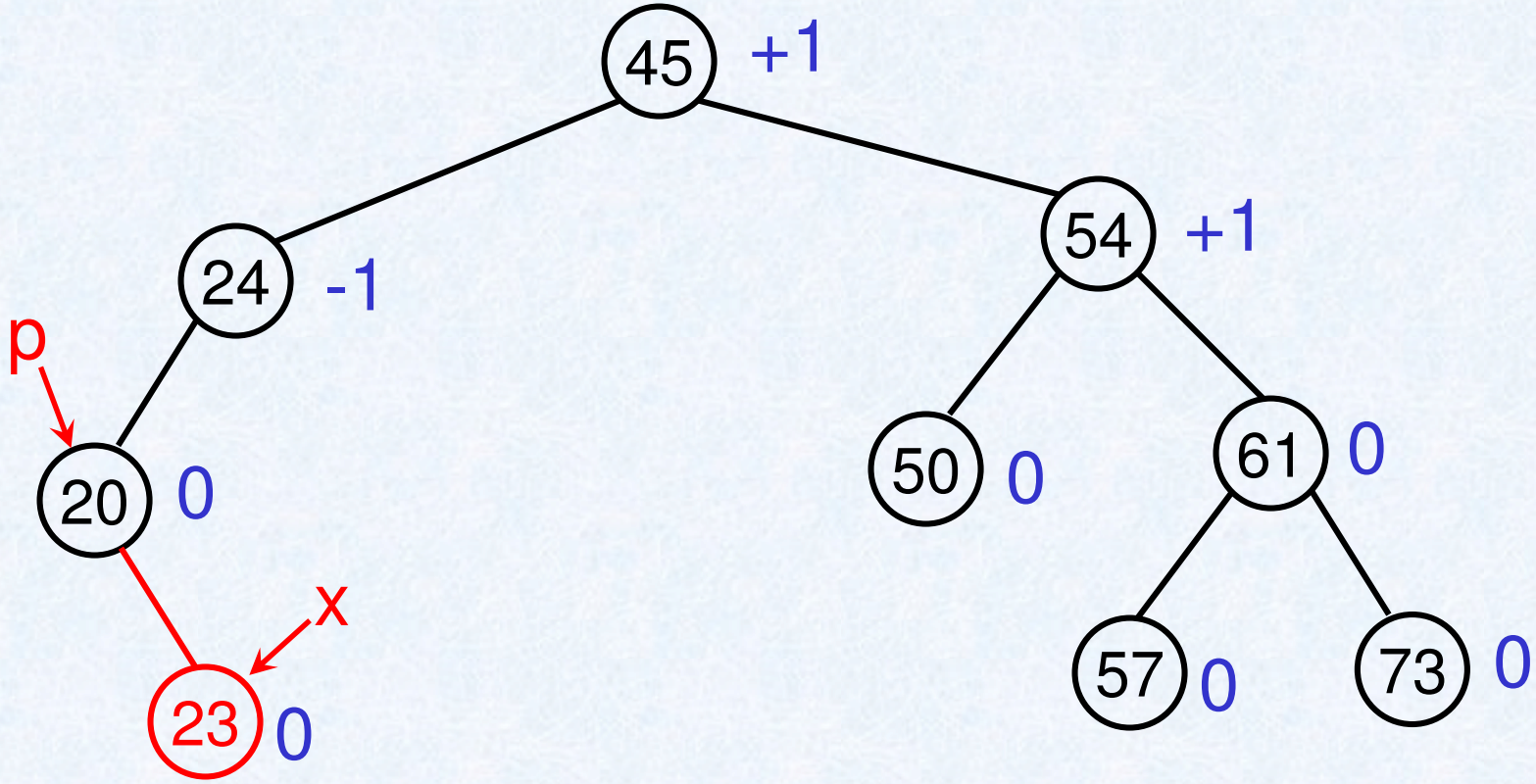
Коррекция узлов



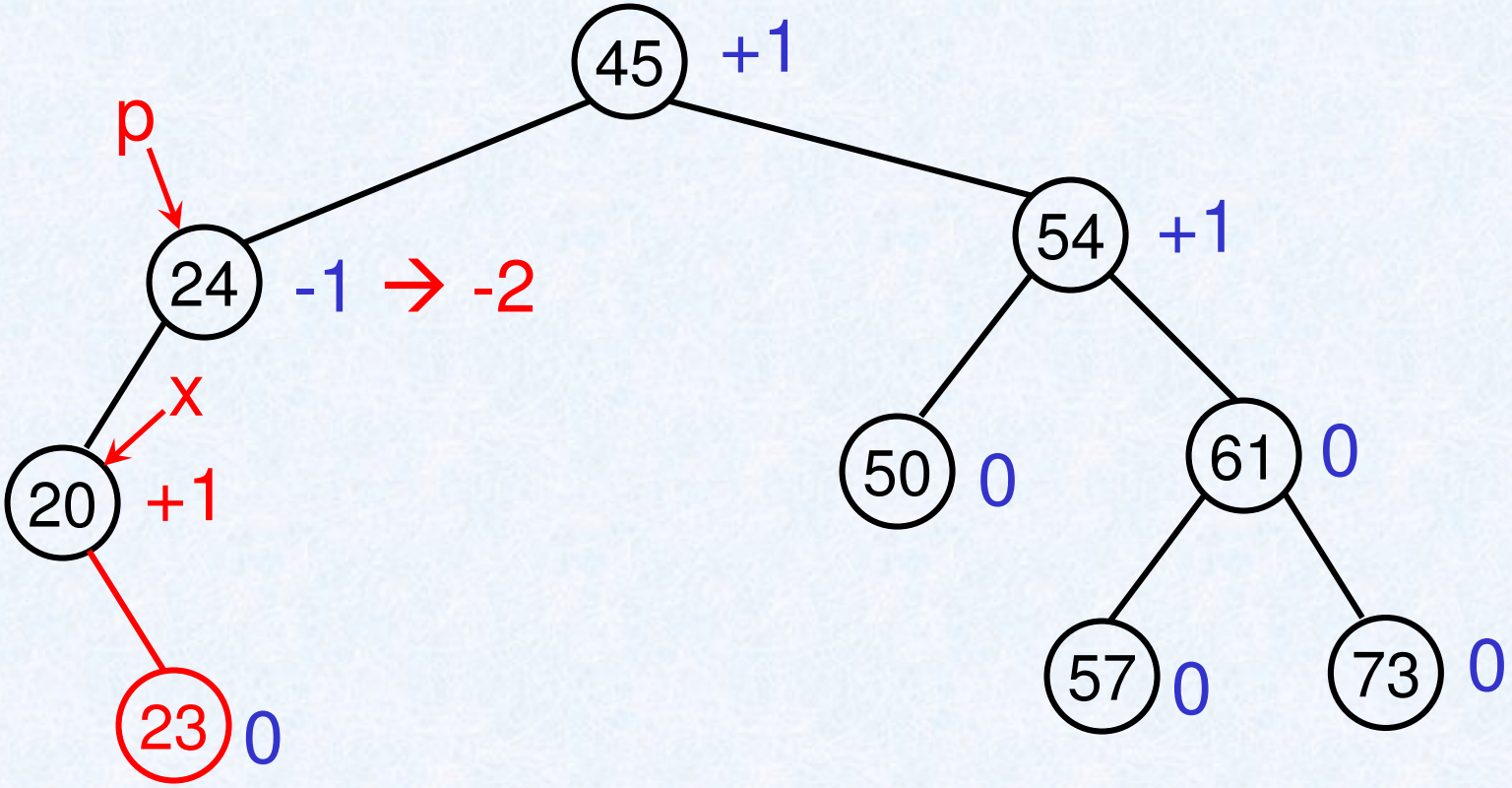
Коррекция узлов



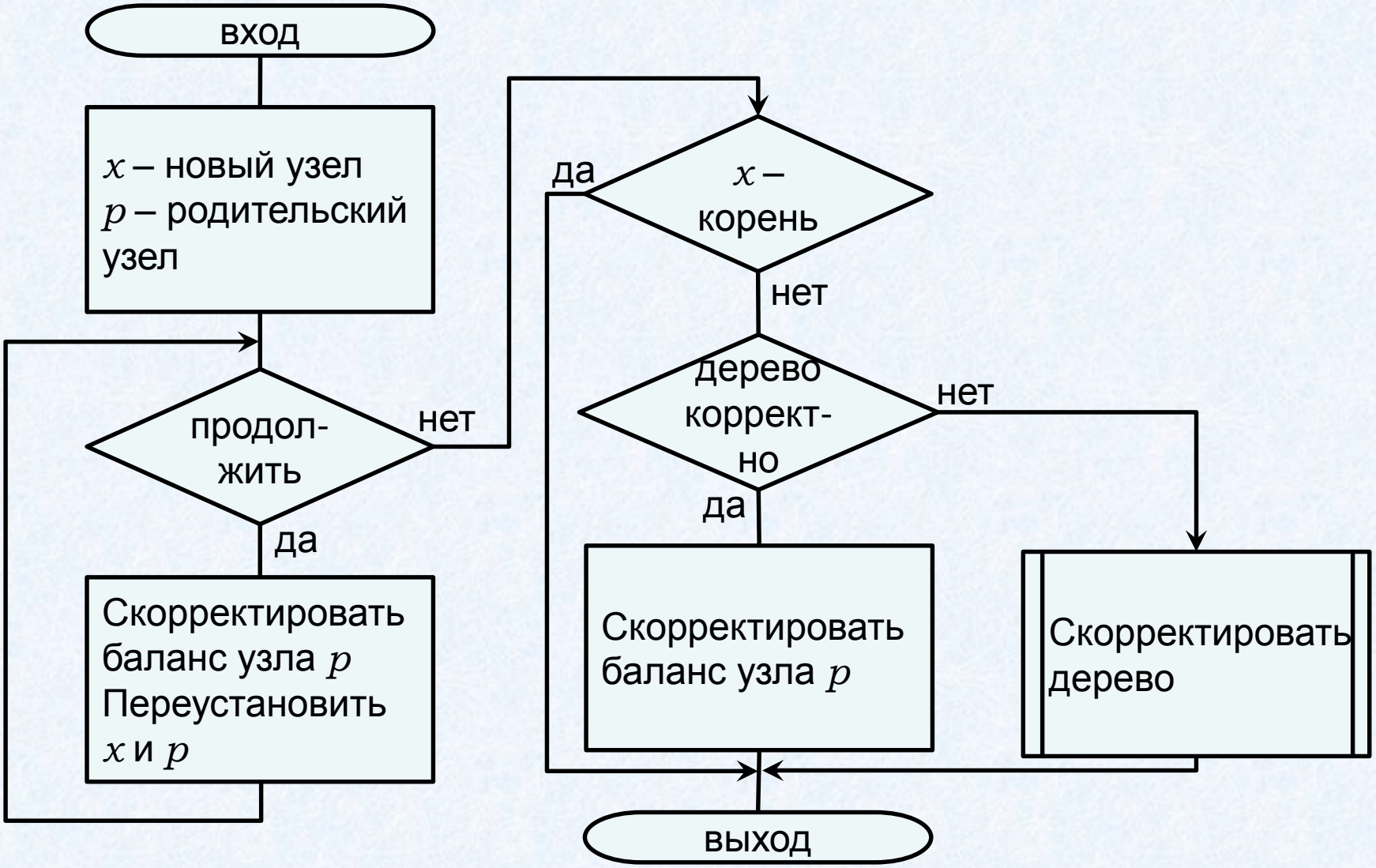
Коррекция узлов



Коррекция узлов



Коррекция узлов



Коррекция узлов

x – новый узел дерева

p = x->parent – родительский узел

```
while p ≠ EList и p->balance = 0 {
   if x в левом поддереве p
      p->balance = -1
   else
      p->balance = +1
   x = p
   p = x->parent
}
```

Коррекция узлов

p – *EList* или узел дерева, у которого
p->*balance* ≠ 0

if p == *EList*

успех

if x в левом поддереве p и p->*balance* = +1 или
x в правом поддереве p и p->*balance* = -1 {

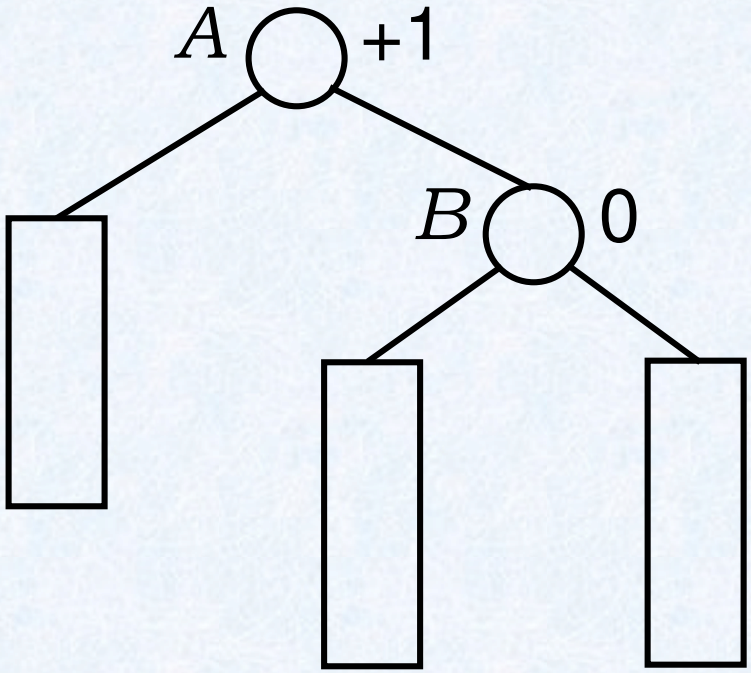
p->*balance* = 0

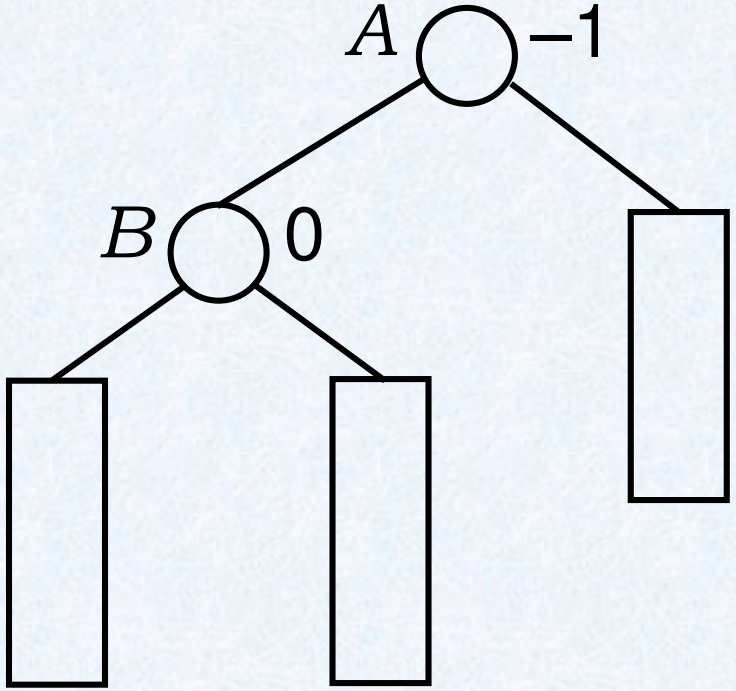
успех

}

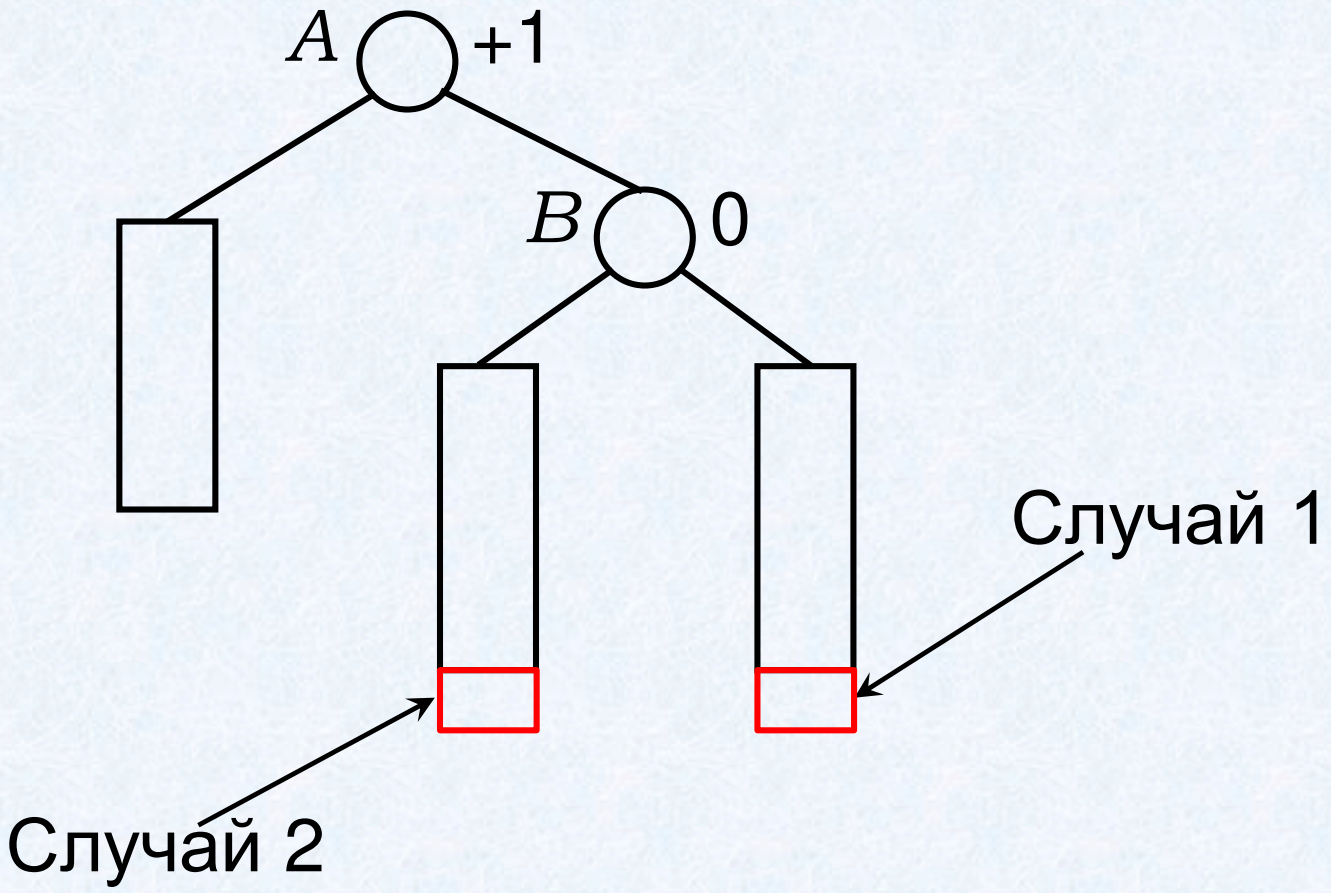
Коррекция дерева

Коррекция AVL- дерева



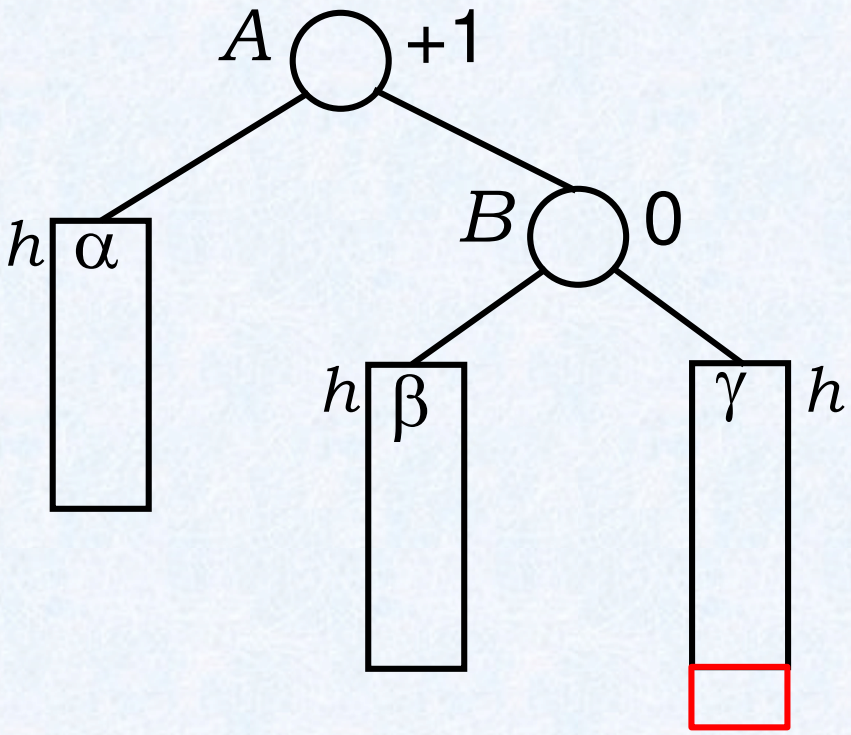


Коррекция AVL- дерева



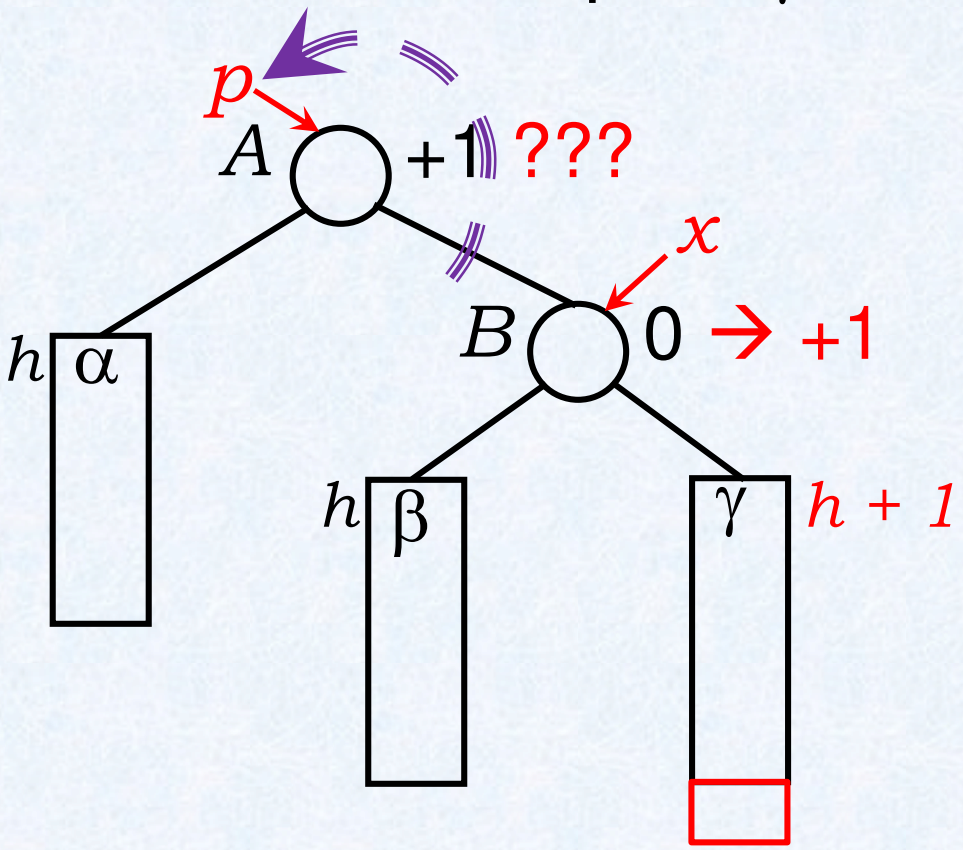
Случай 1

Вставка в поддерево γ



Случай 1

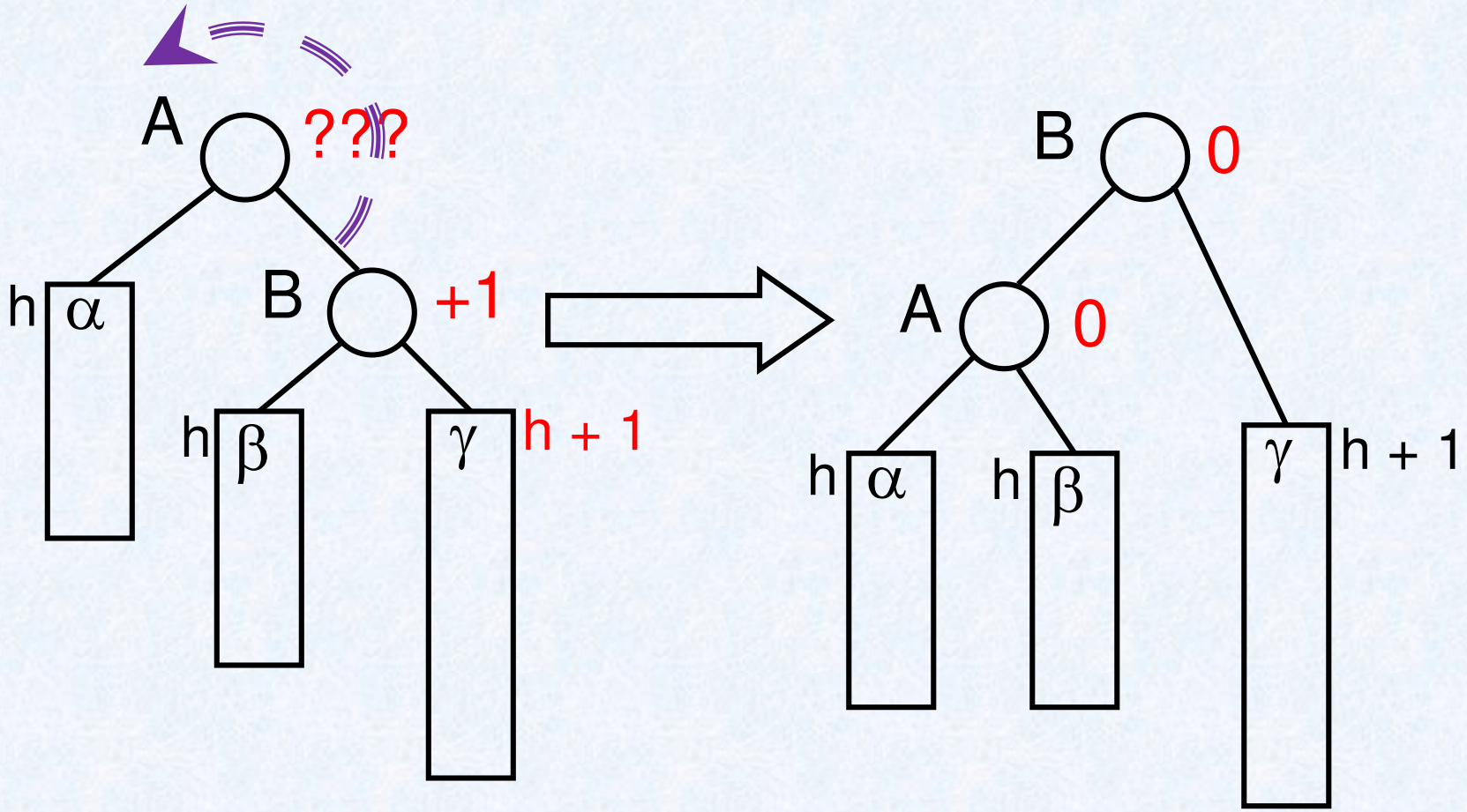
Вставка в поддерево γ



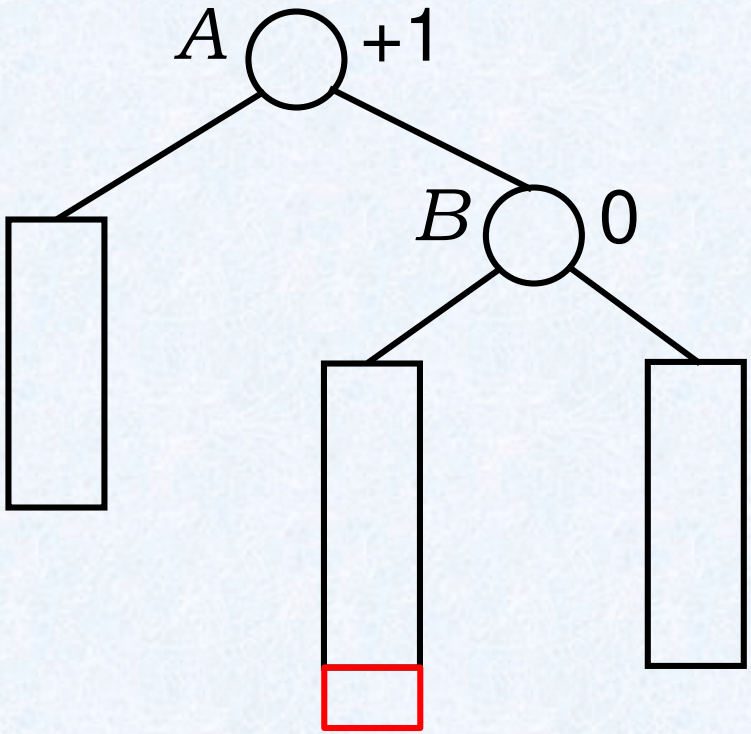
Решение:

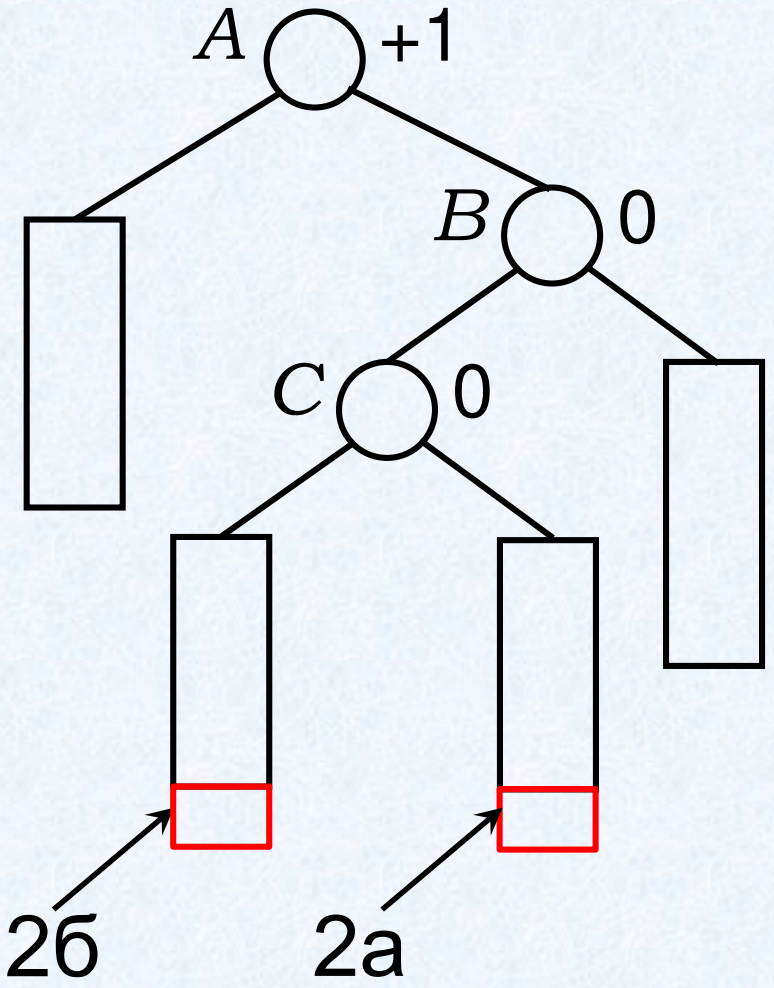
левый поворот вокруг узла A

Случай 1



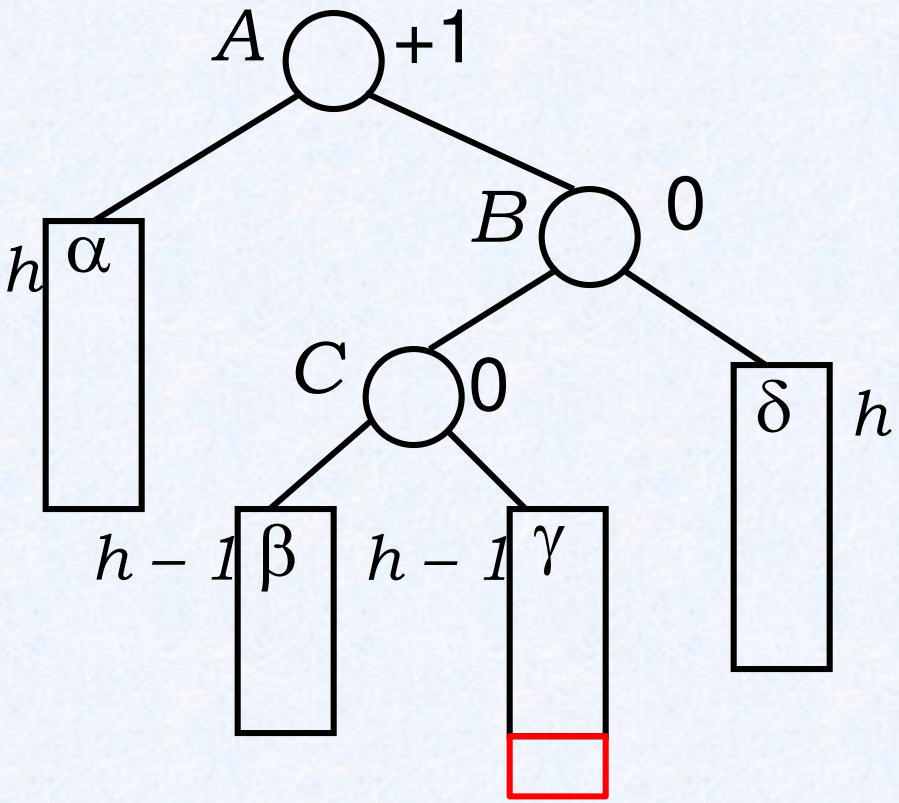
Случай 2





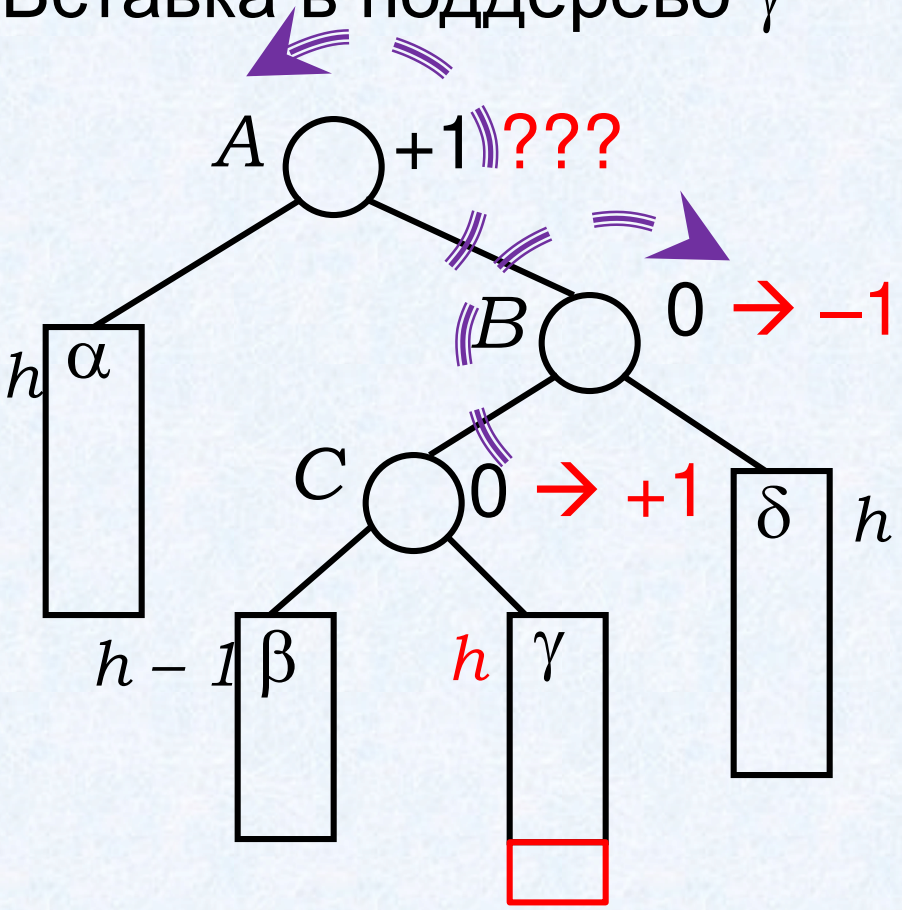
Случай 2а

Вставка в поддерево γ



Случай 2а

Вставка в поддерево γ



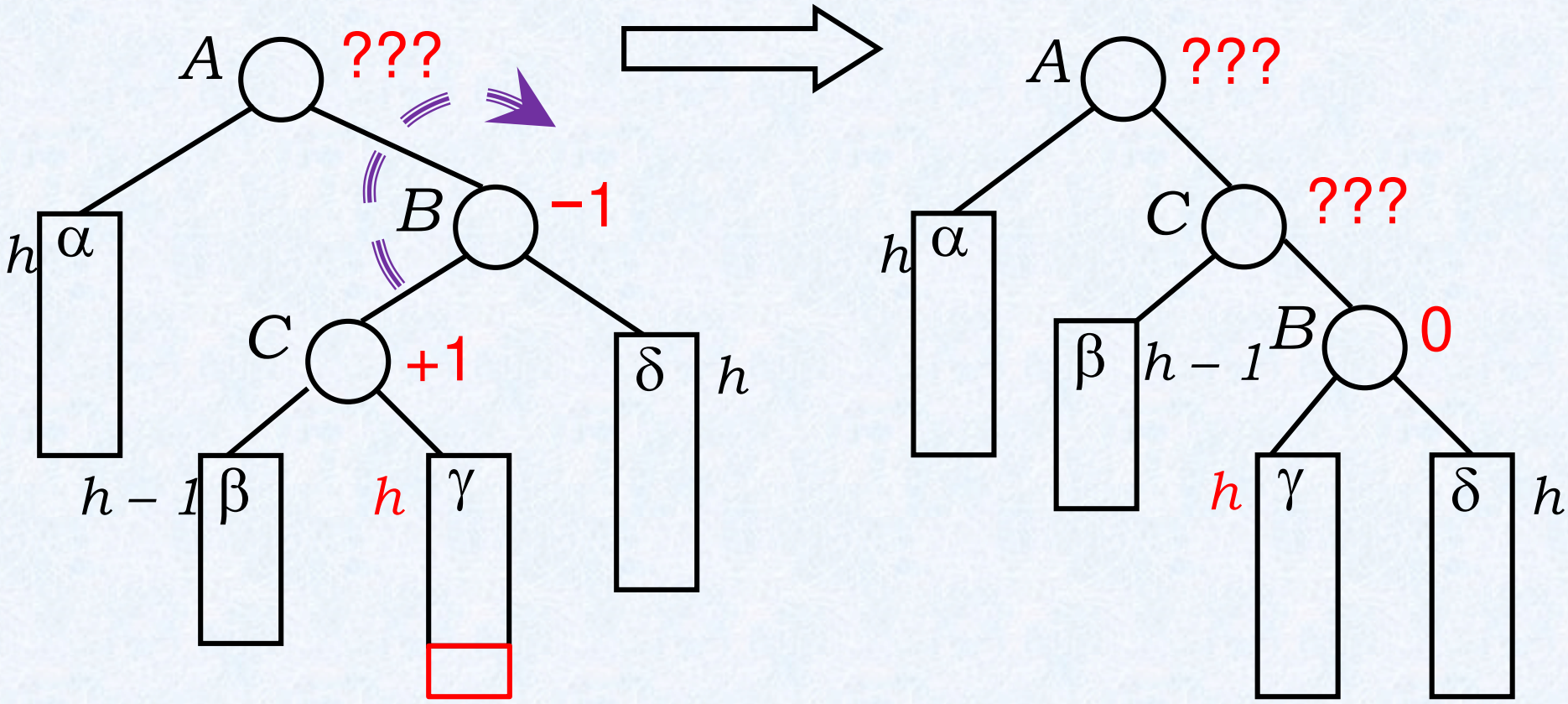
Решение:

Шаг 1 – правый поворот вокруг узла B

Шаг 2 – левый поворот вокруг узла A

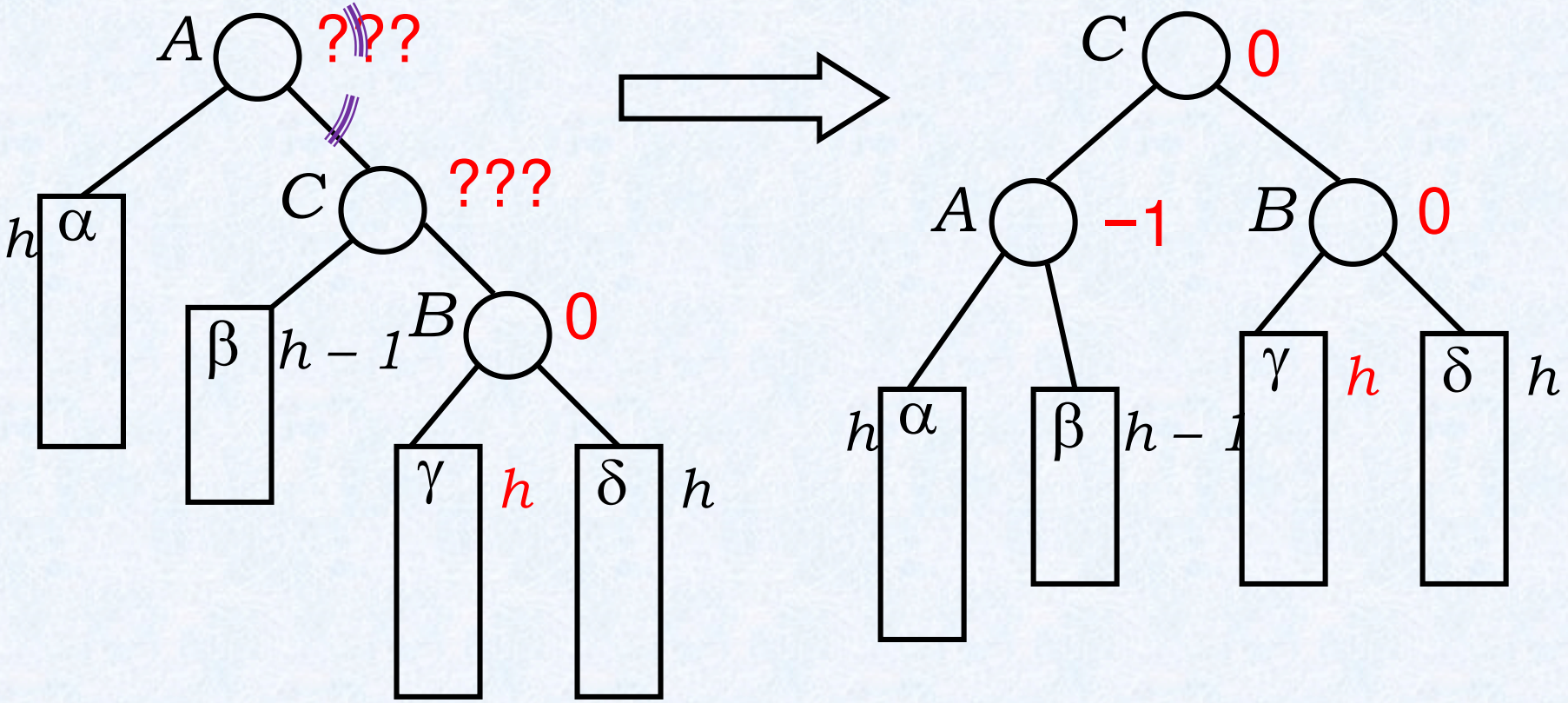
Случай 2а

Шаг 1



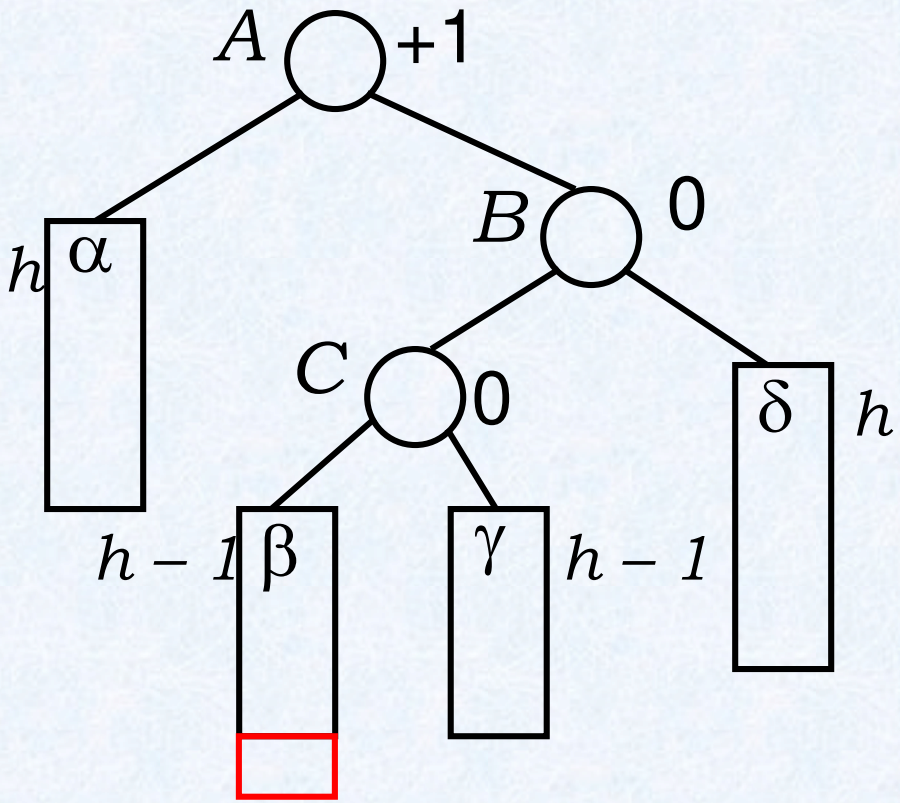
Случай 2а

Шаг 2



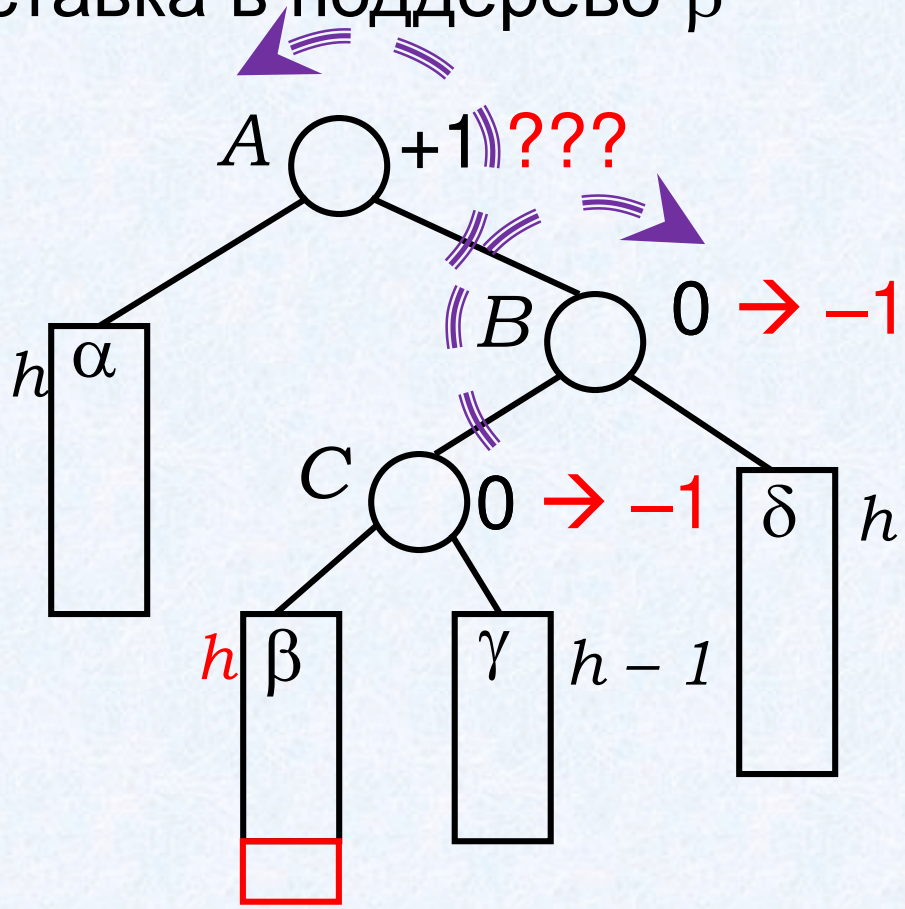
Случай 2б

Вставка в поддерево β



Случай 2б

Вставка в поддерево β



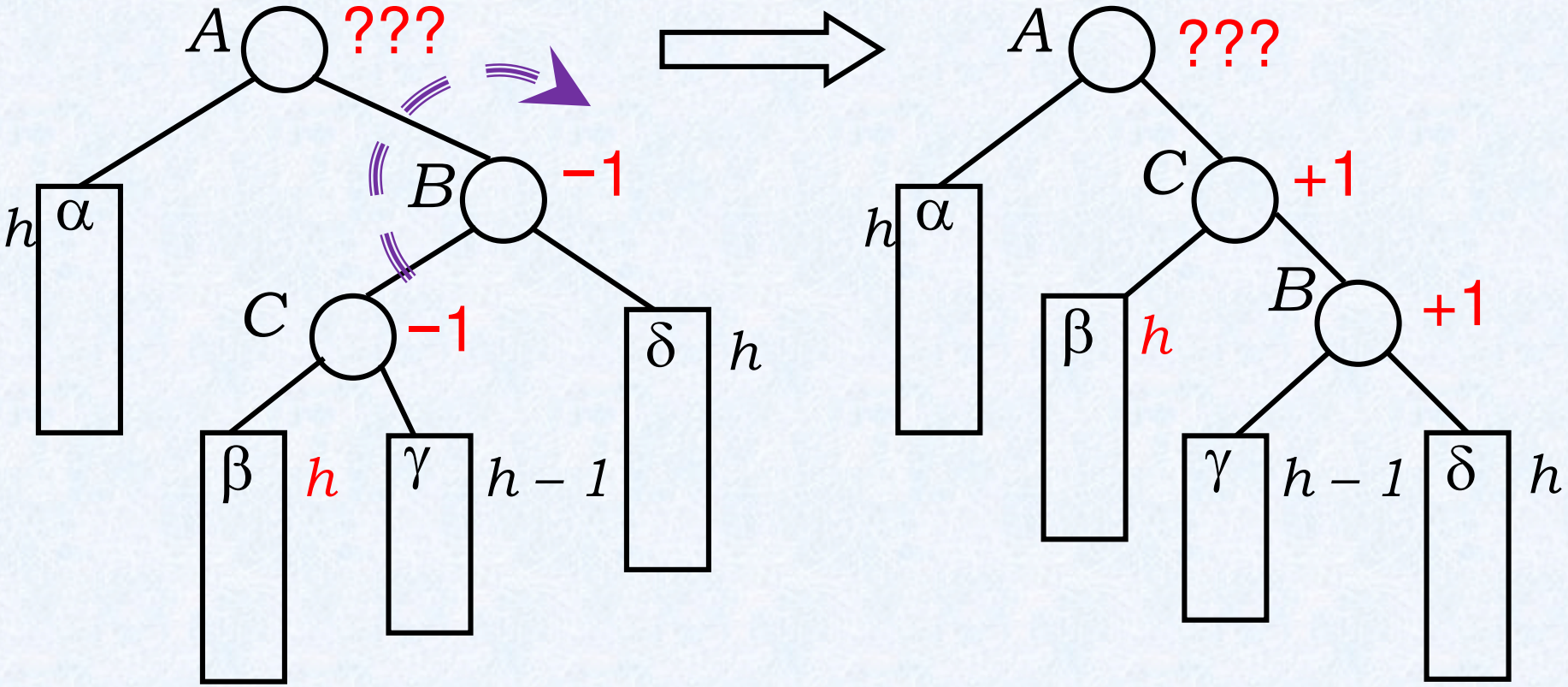
Решение:

Шаг 1 – правый поворот вокруг узла B

Шаг 2 – левый поворот вокруг узла A

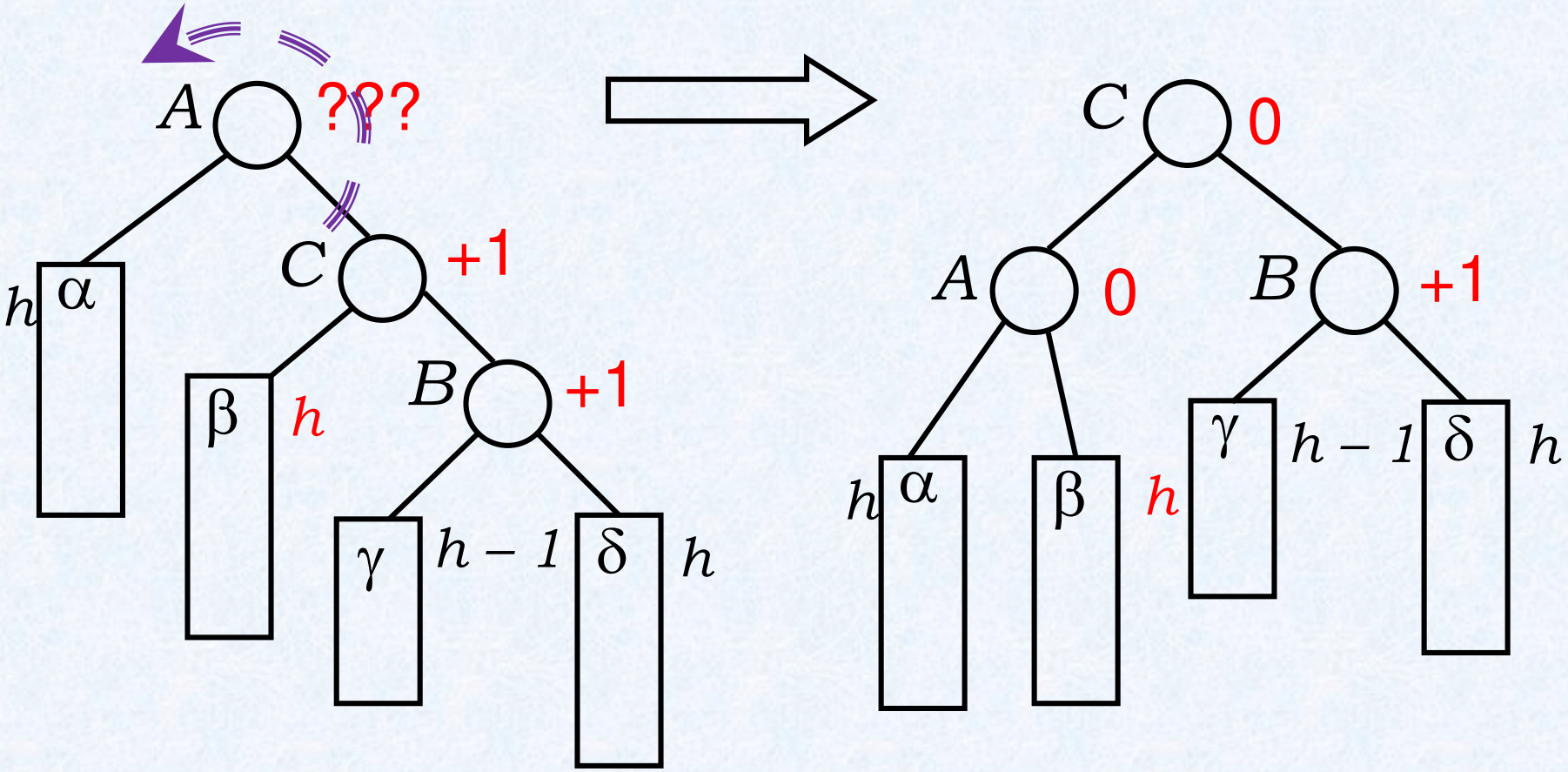
Случай 2б

Шаг 1



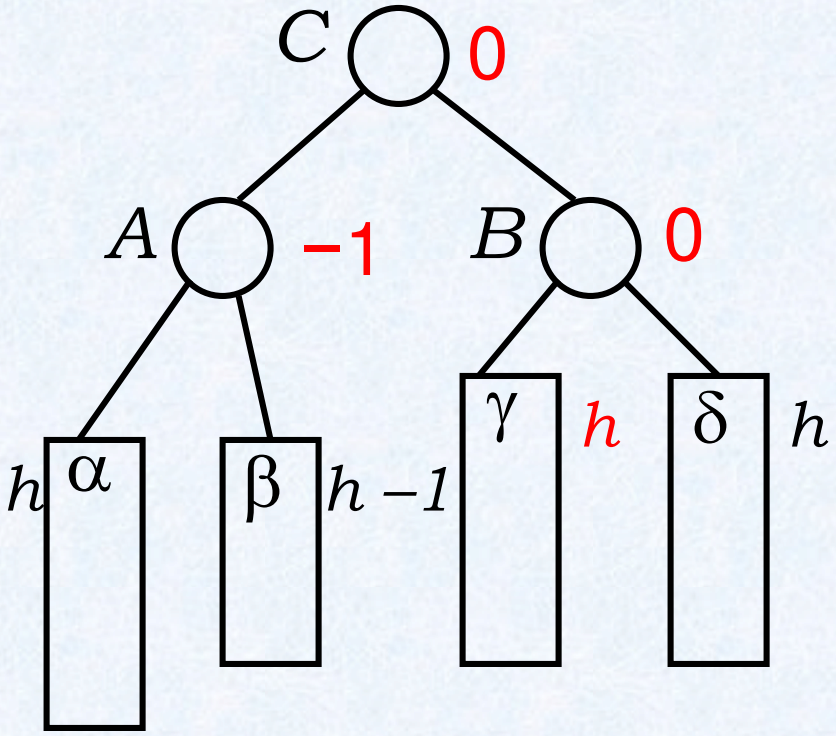
Случай 2б

Шаг 2

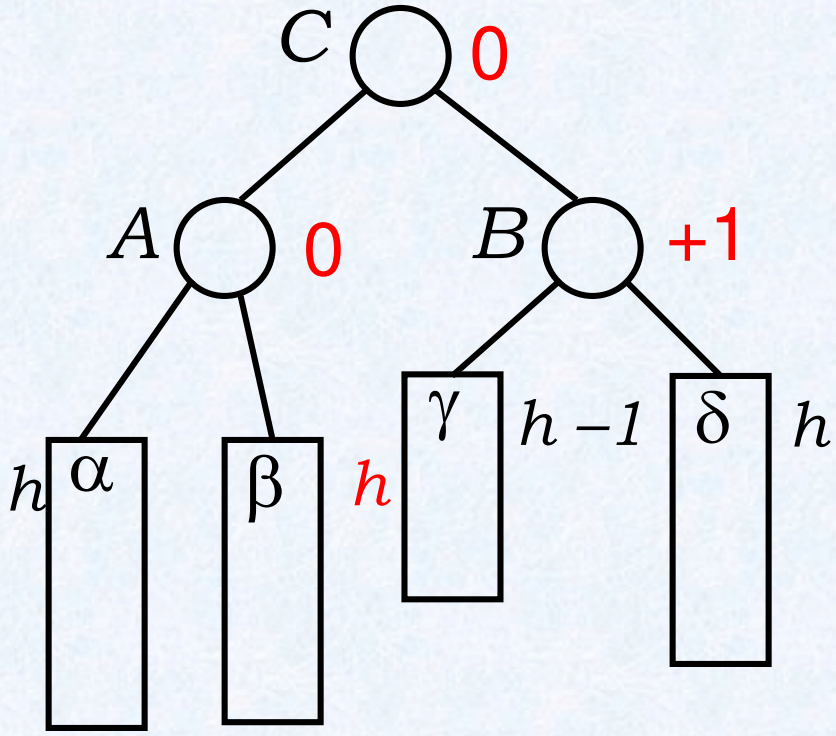


Случаи 2а и 2б

Случай 2а



Случай 2б



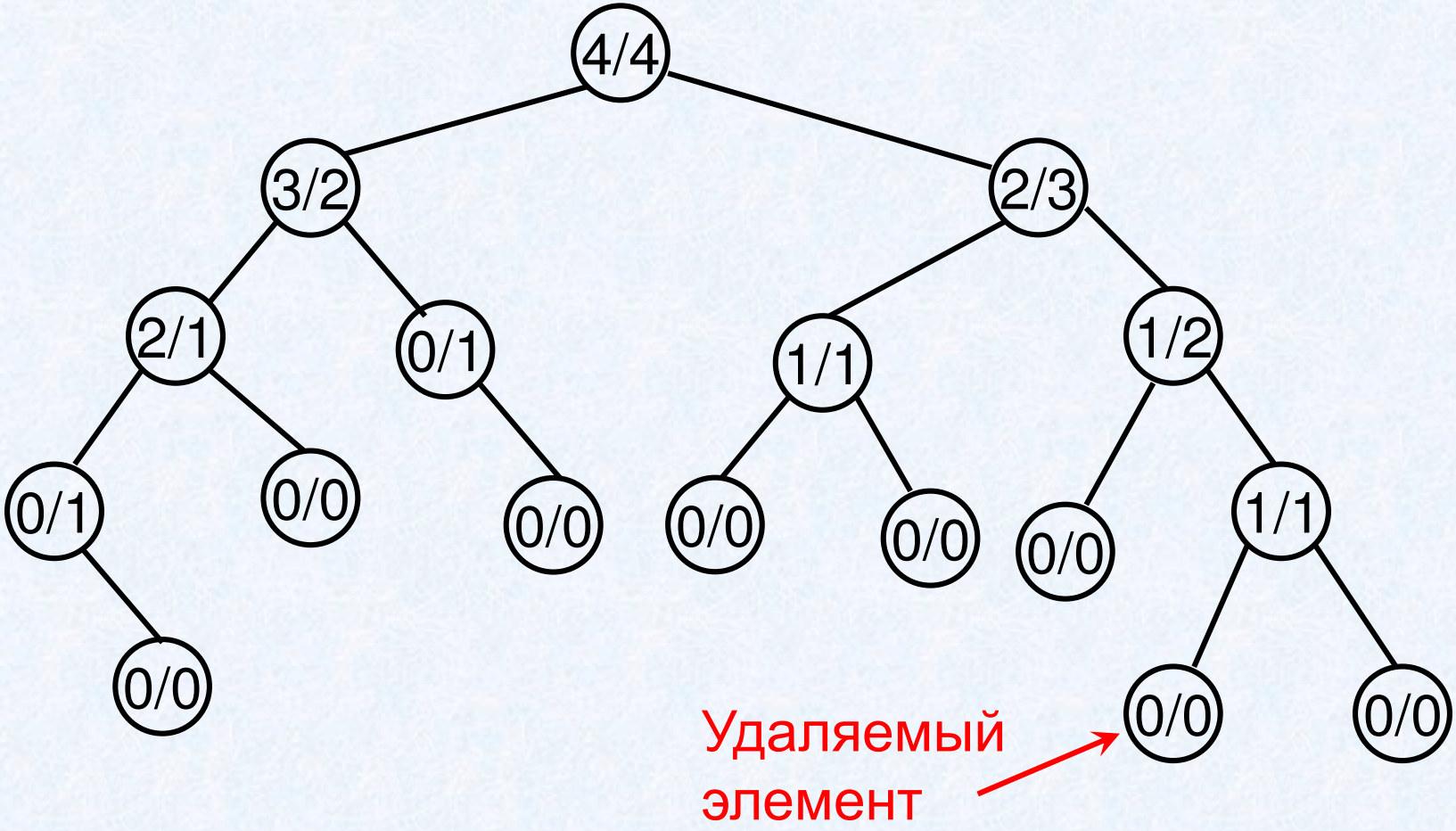
Удаление из AVL-дерева

- Удалить элемент так же, как в двоичном дереве поиска (с учетом внешних листьев – *EList*)
- Скорректировать свойство сбалансированности узлов дерева
- При необходимости скорректировать само AVL-дерево

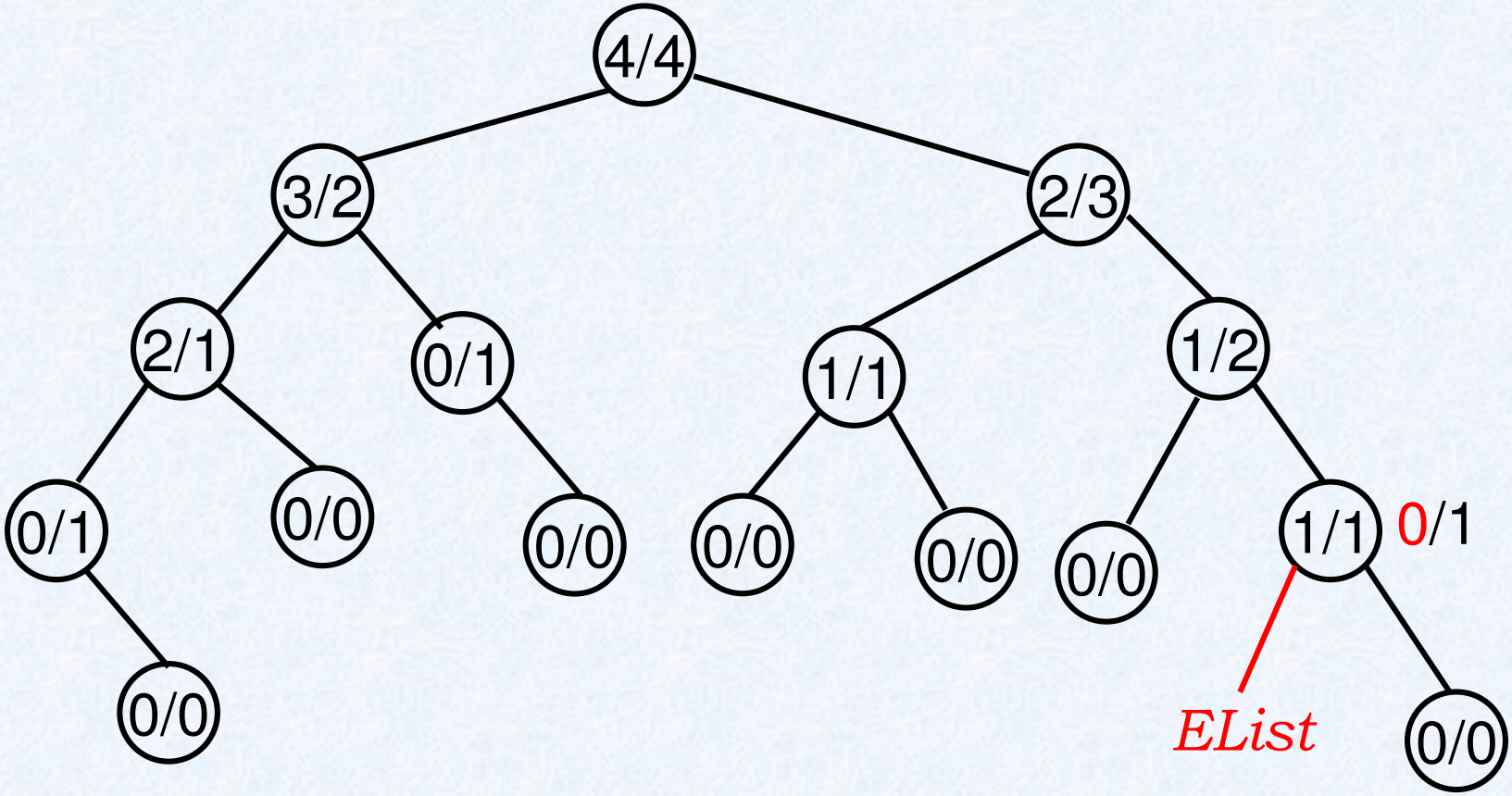
Узел AVL-дерева

```
struct Node {
   int key;
   int hleft; – высота левого поддерева
   int hright; – высота правого поддерева
   struct Node *left, *right, *parent;
};
```

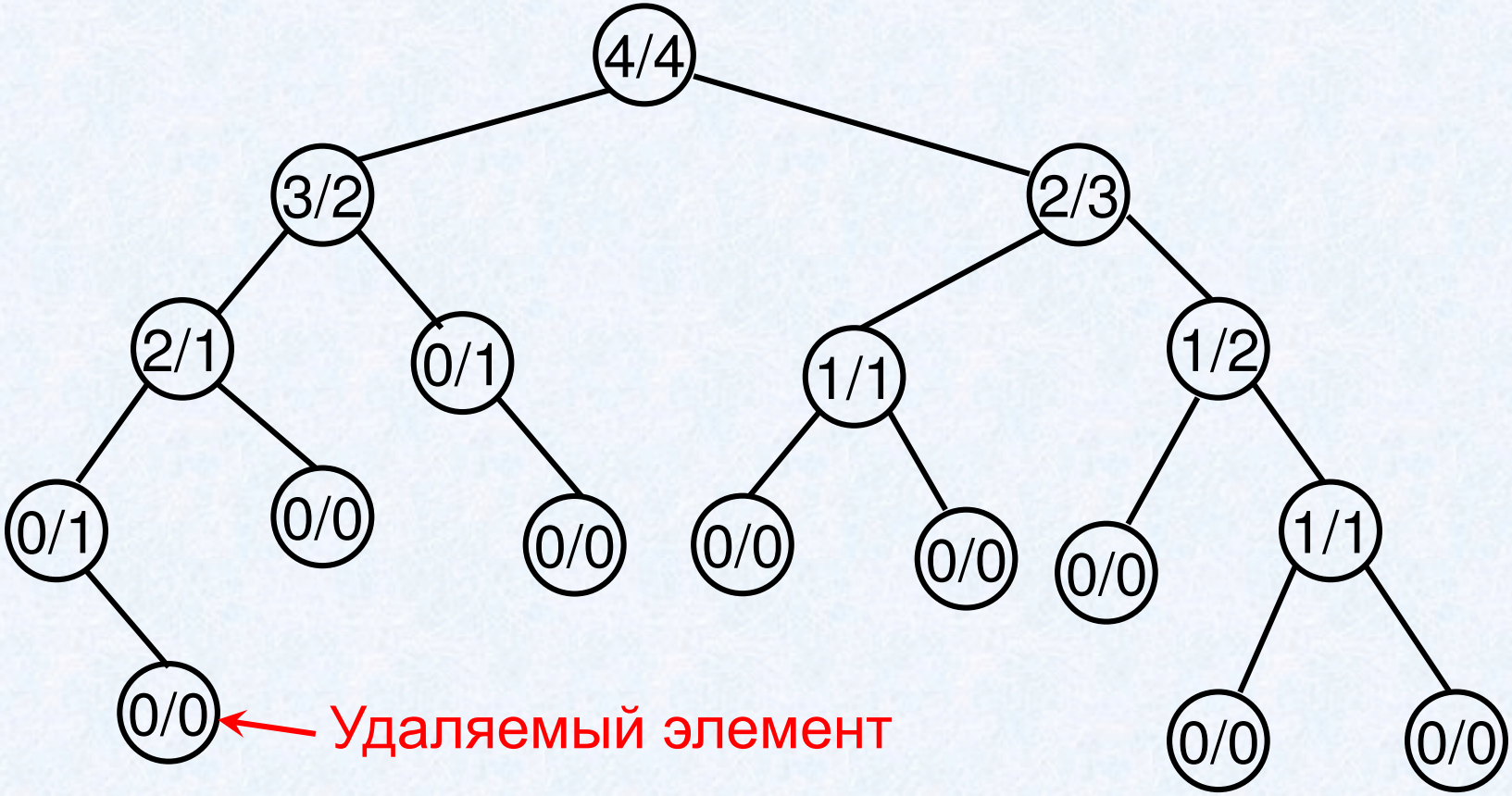
Примеры удаления из AVL-дерева



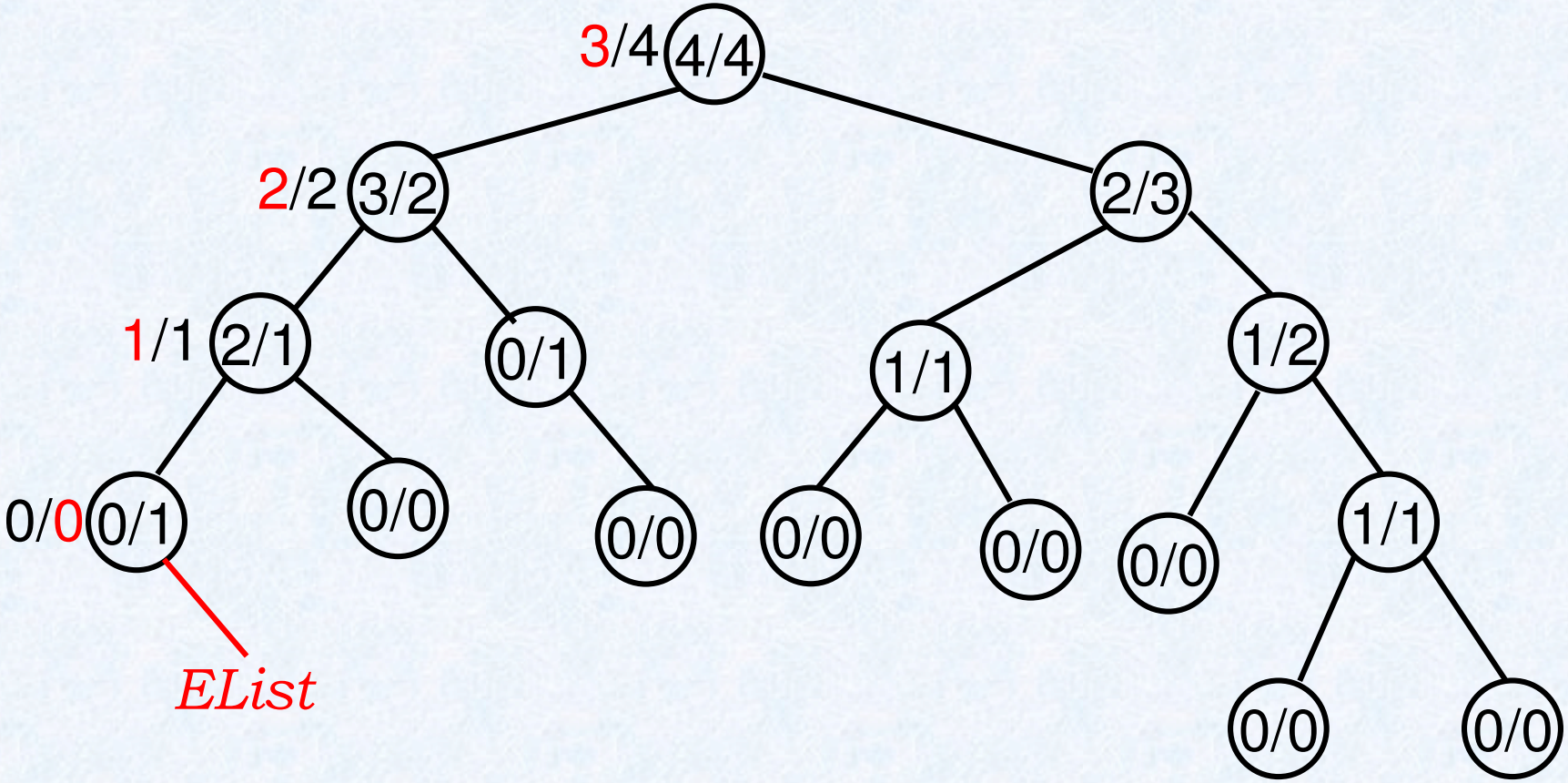
Примеры удаления из AVL-дерева



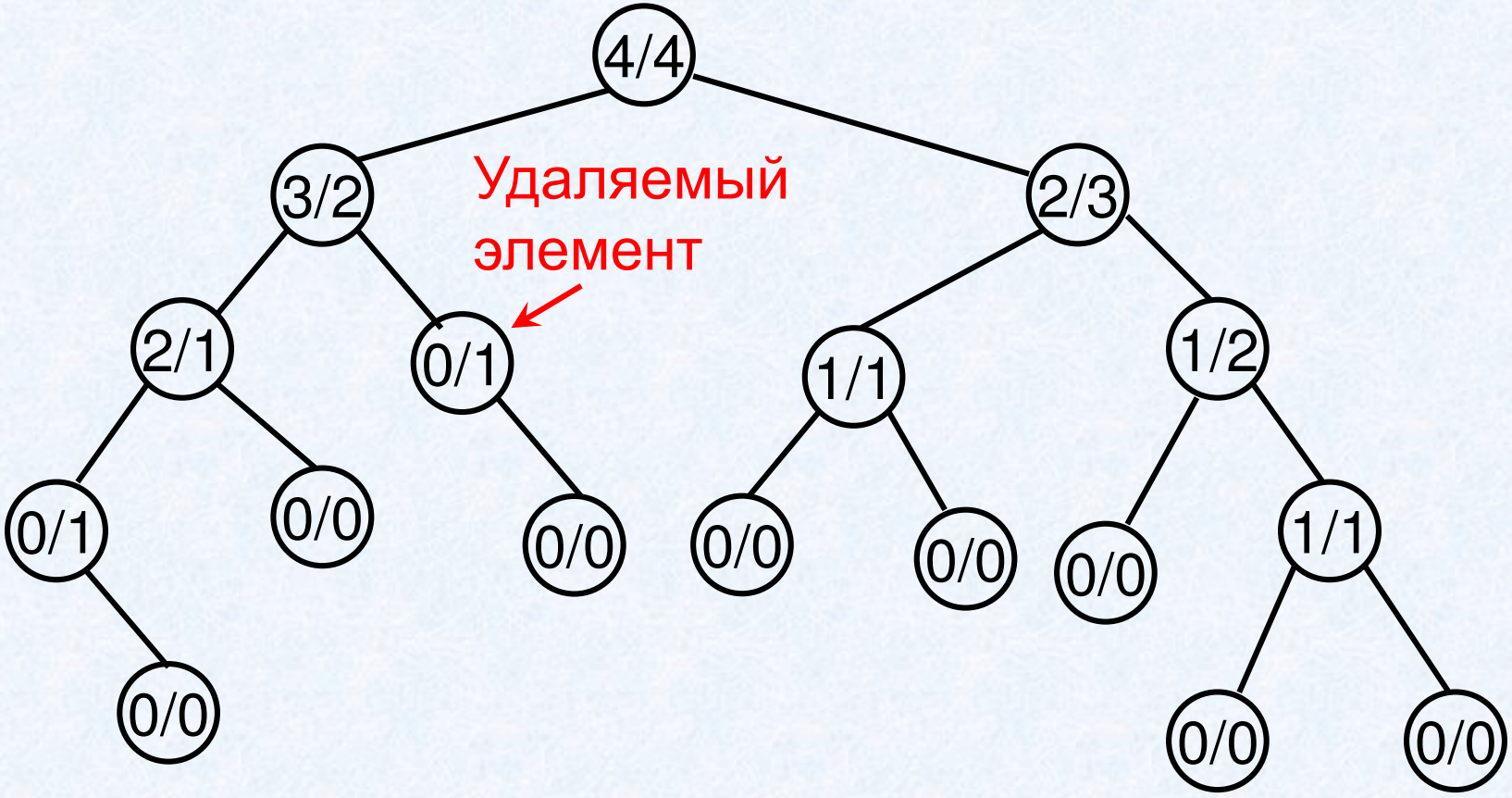
Примеры удаления из AVL-дерева



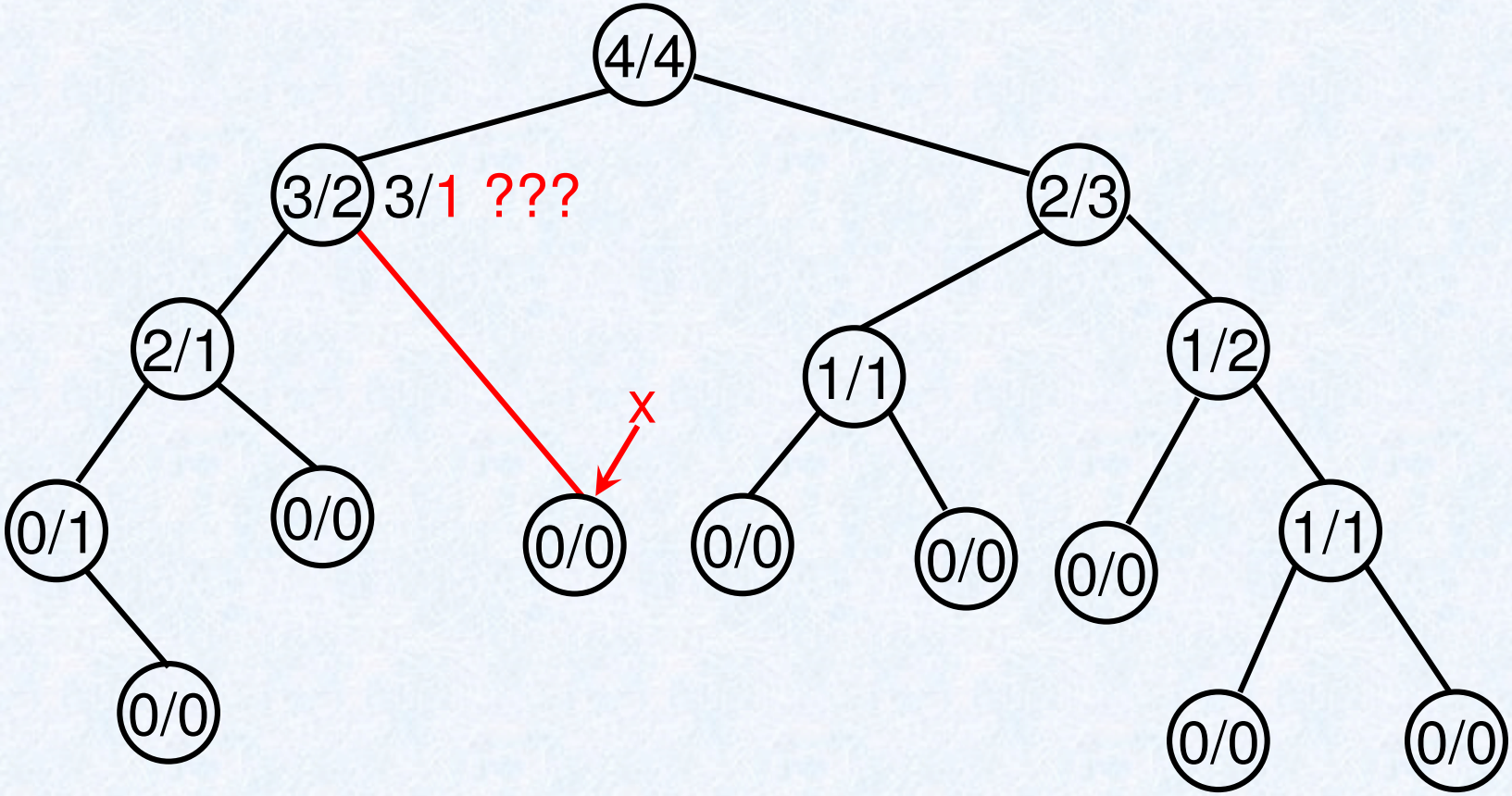
Примеры удаления из AVL-дерева



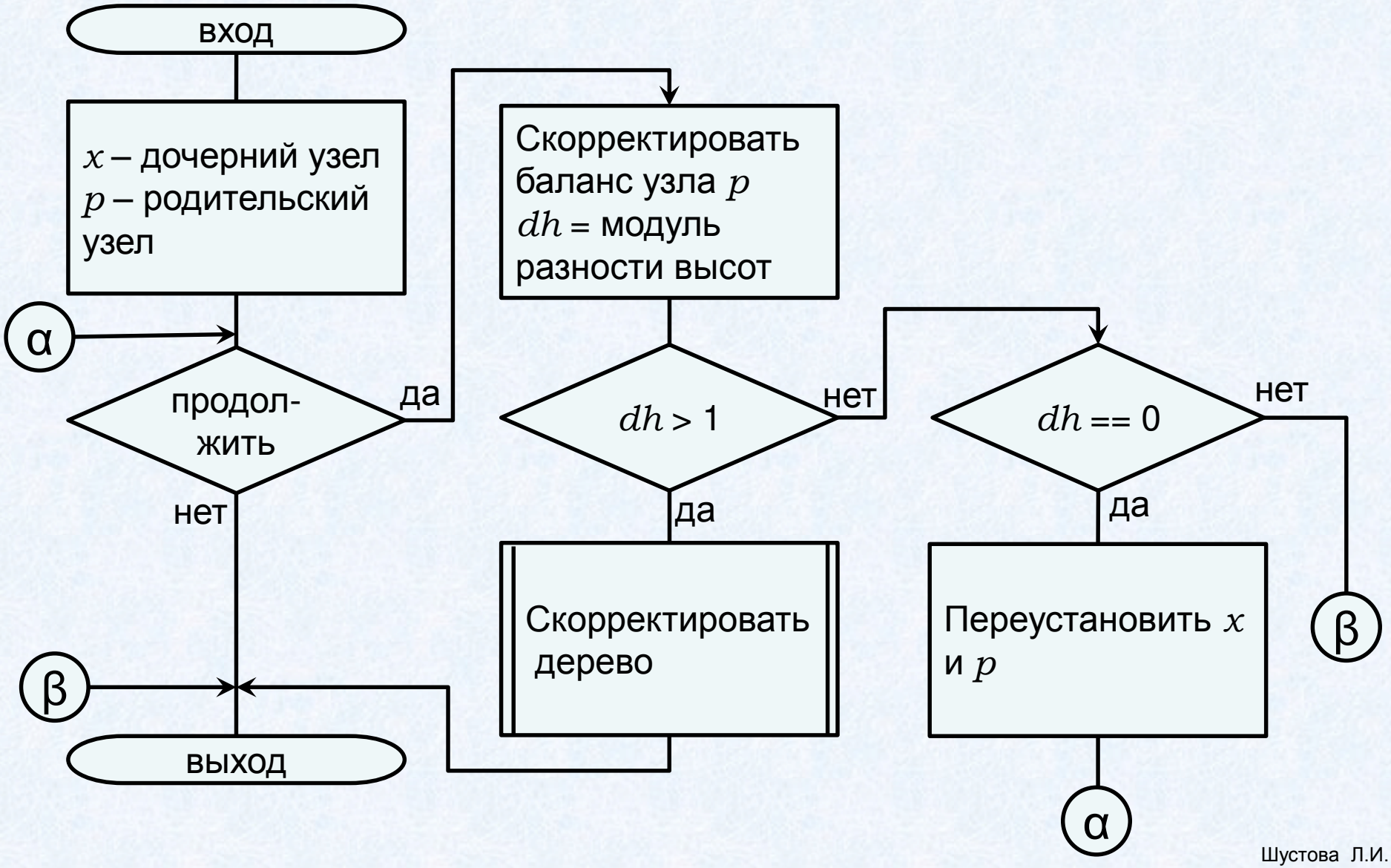
Примеры удаления из AVL-дерева



Примеры удаления из AVL-дерева



Коррекция узлов



Коррекция узлов

Удалить узел из бинарного дерева поиска

x – дочерний узел удаленного элемента

$p = x\text{->}parent$

while $p \neq EList$ {

коррекция узла дерева p:

if x в левом поддереве p

$p\text{->}hleft = p\text{->}hleft - 1$

else

$p\text{->}hright = p\text{->}hright - 1$

Коррекция узлов

```
dh = | p->hleft – p->hright |
if dh > 1 {
   коррекция дерева
   успех
}
else
    if dh == 1
      успех
```

Коррекция узлов

else

x = p

p = x->parent

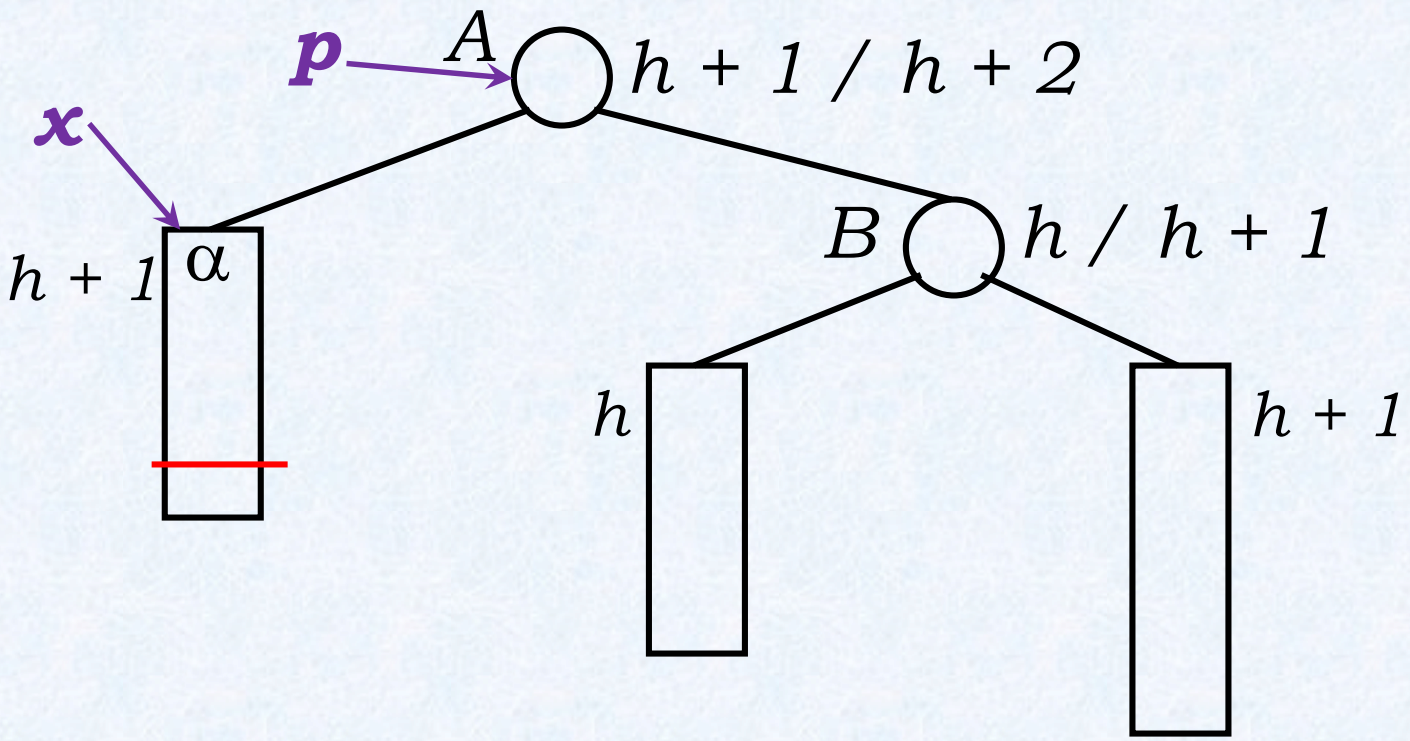
} -- цикл *while*

p – узел AVL-дерева, для которого нарушены требования сбалансированности

Варианты удаления

Удаление из поддерева α

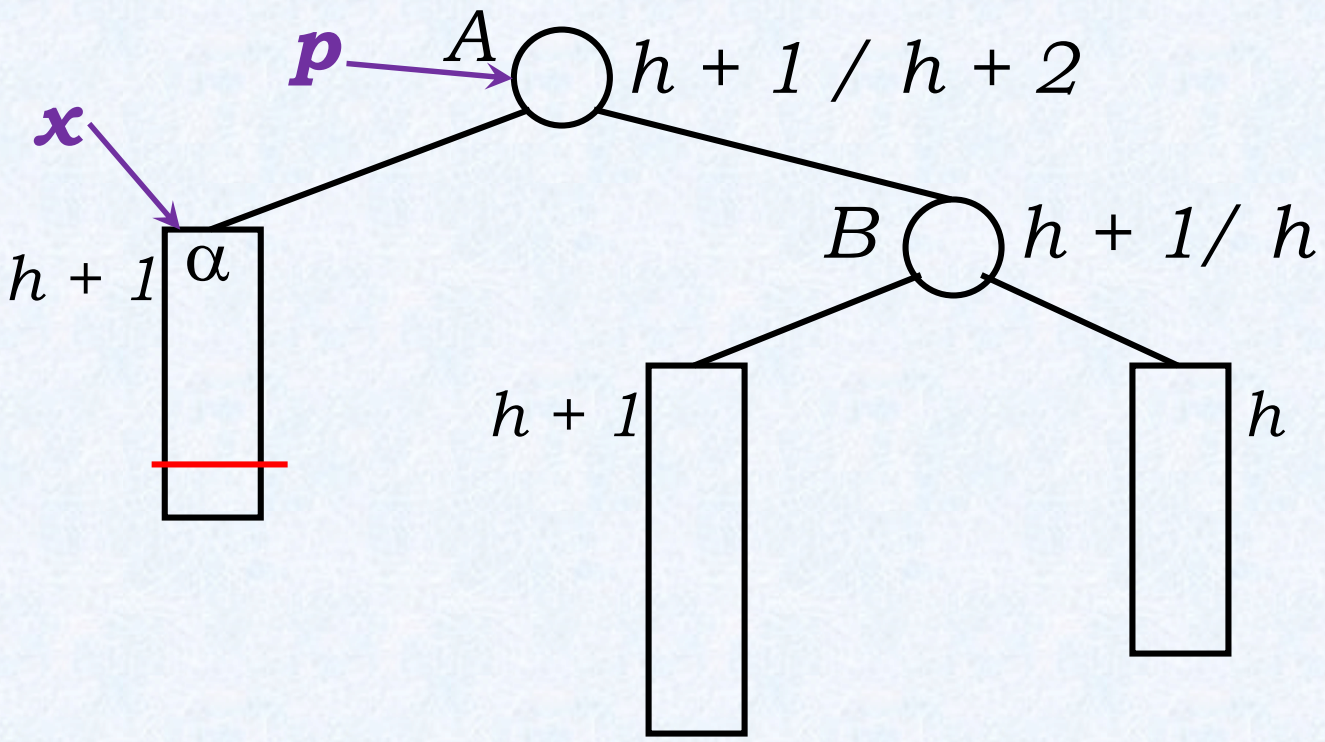
Случай 1



Варианты удаления

Удаление из поддерева α

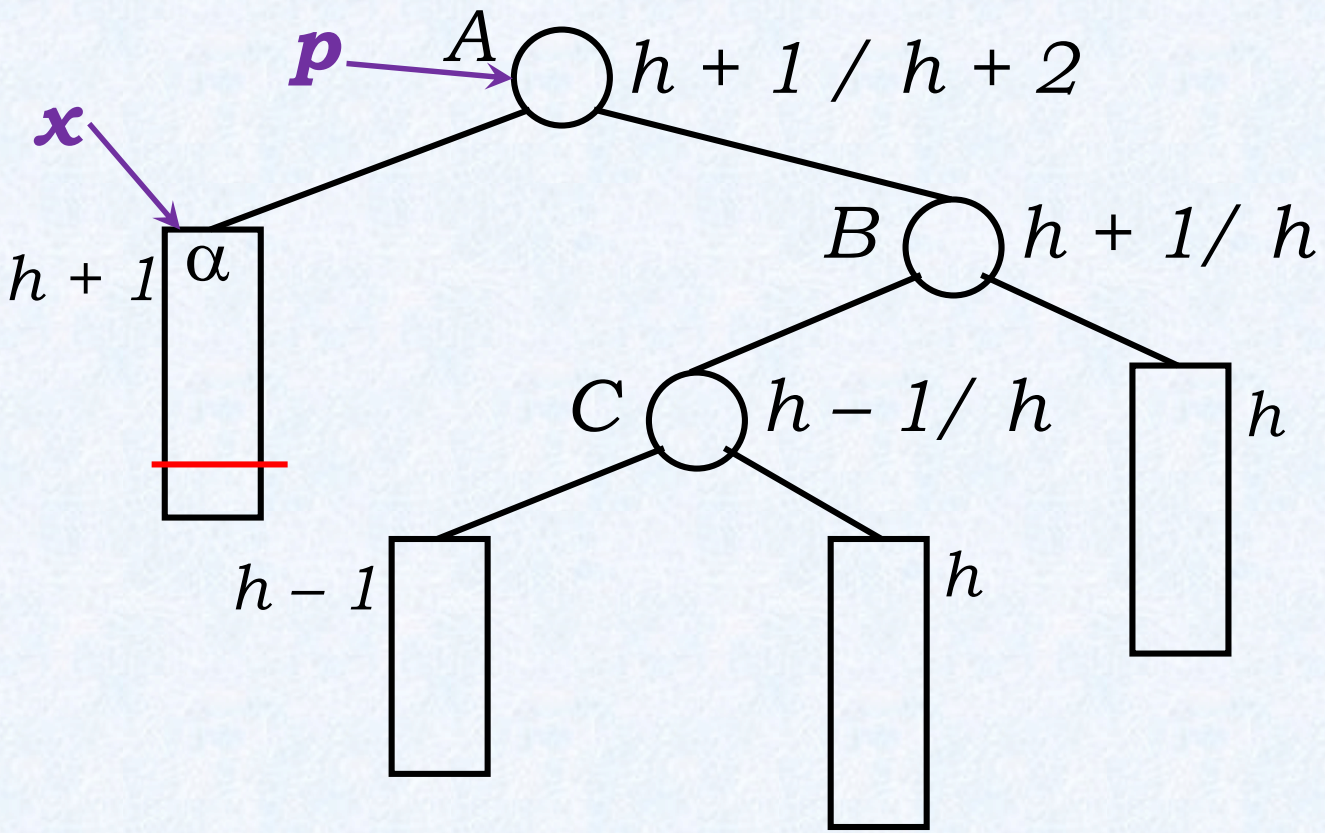
Случай 2



Варианты удаления

Удаление из поддерева α

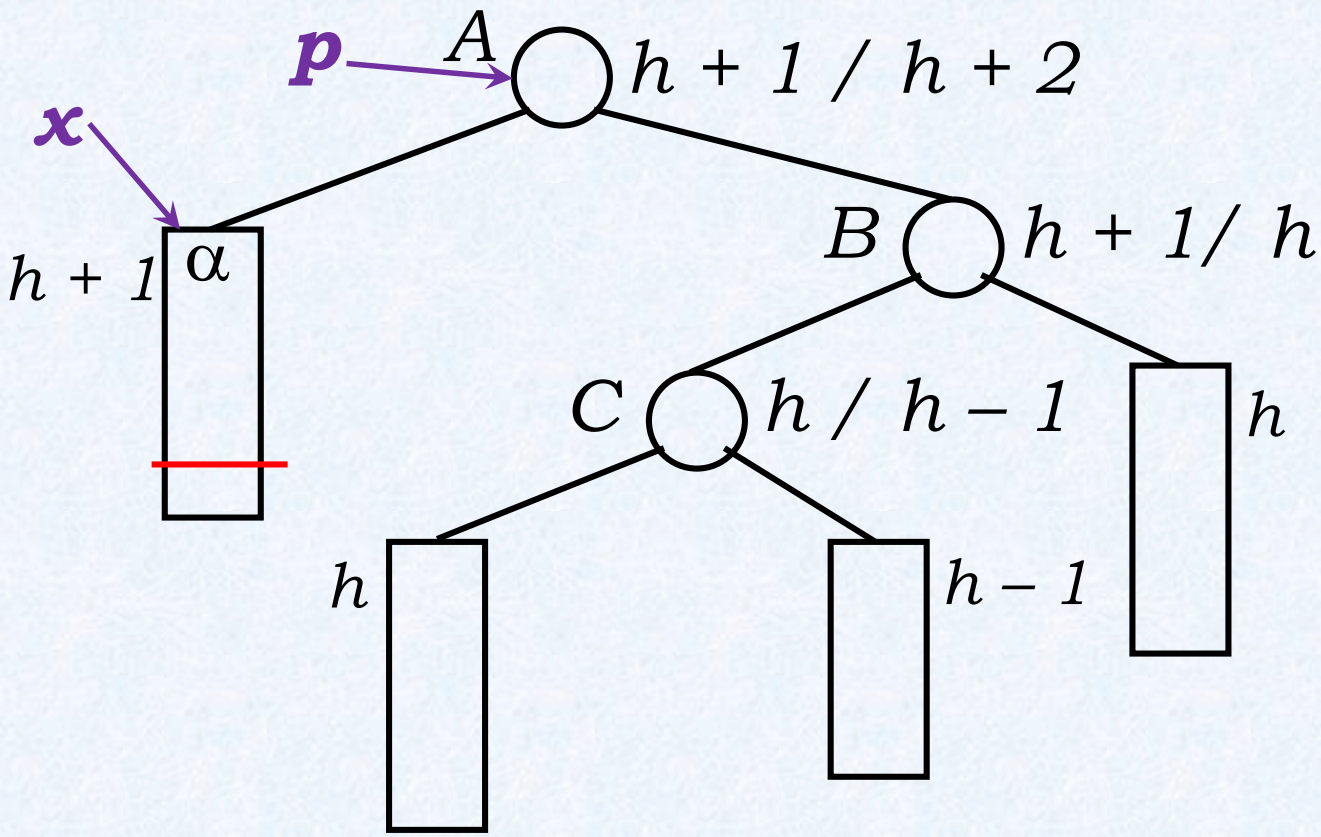
Случай 2а



Варианты удаления

Удаление из поддерева α

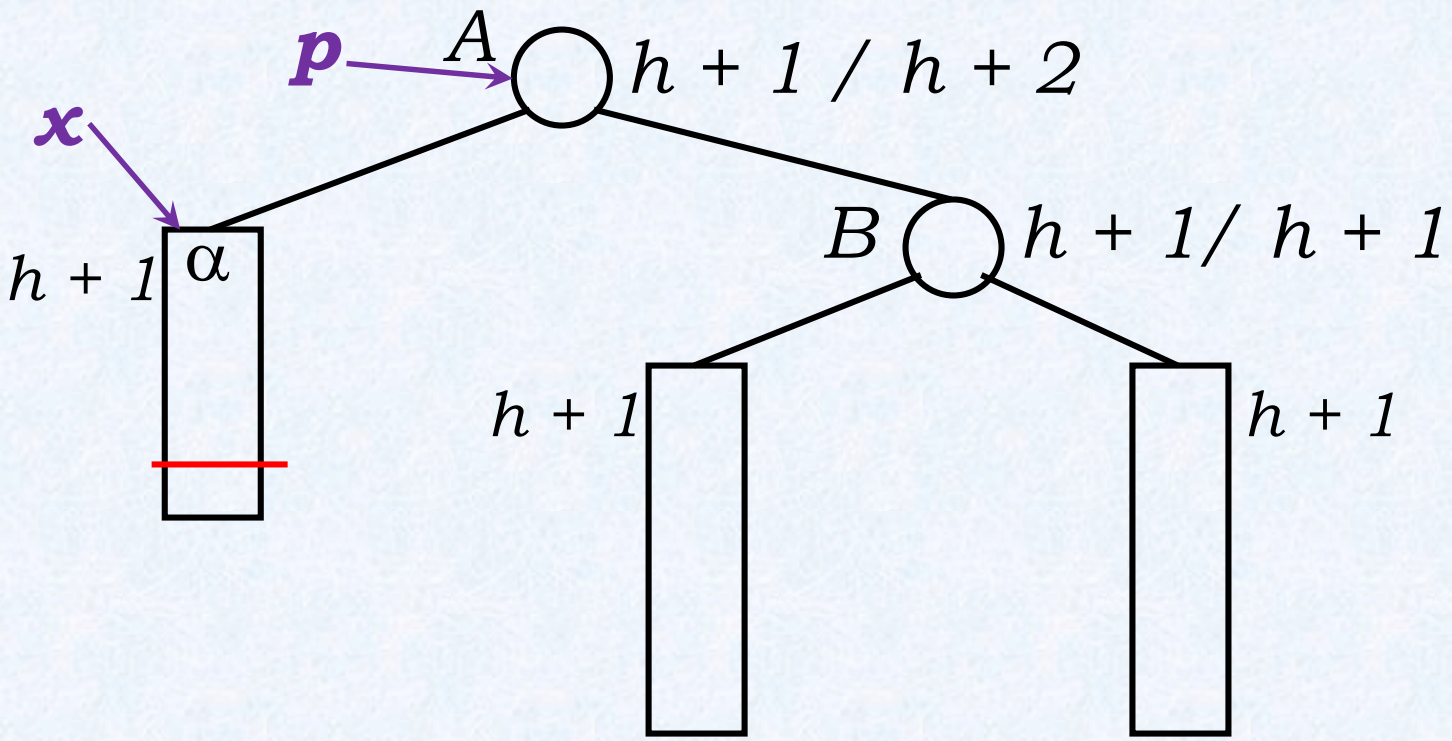
Случай 2б



Варианты удаления

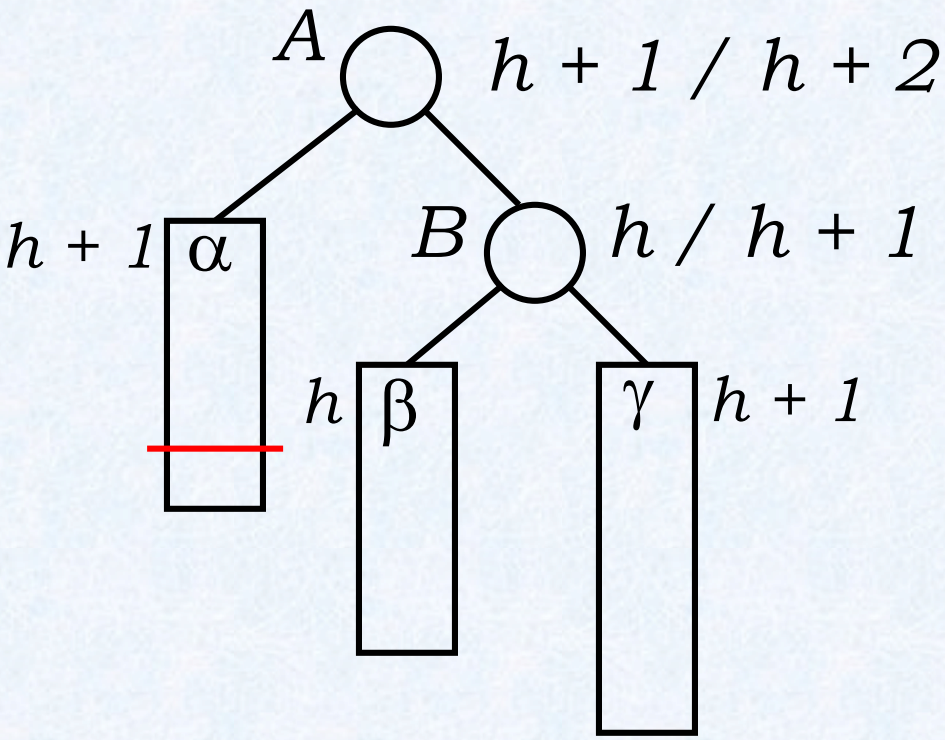
Удаление из поддерева α

Случай 3



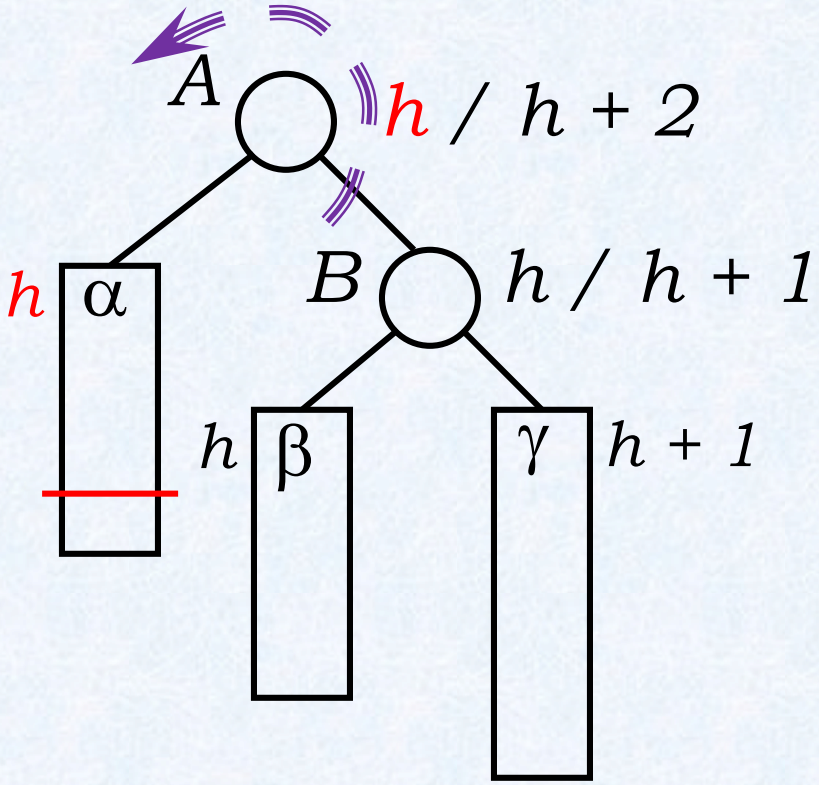
Случай 1

Удаление из поддерева α



Случай 1

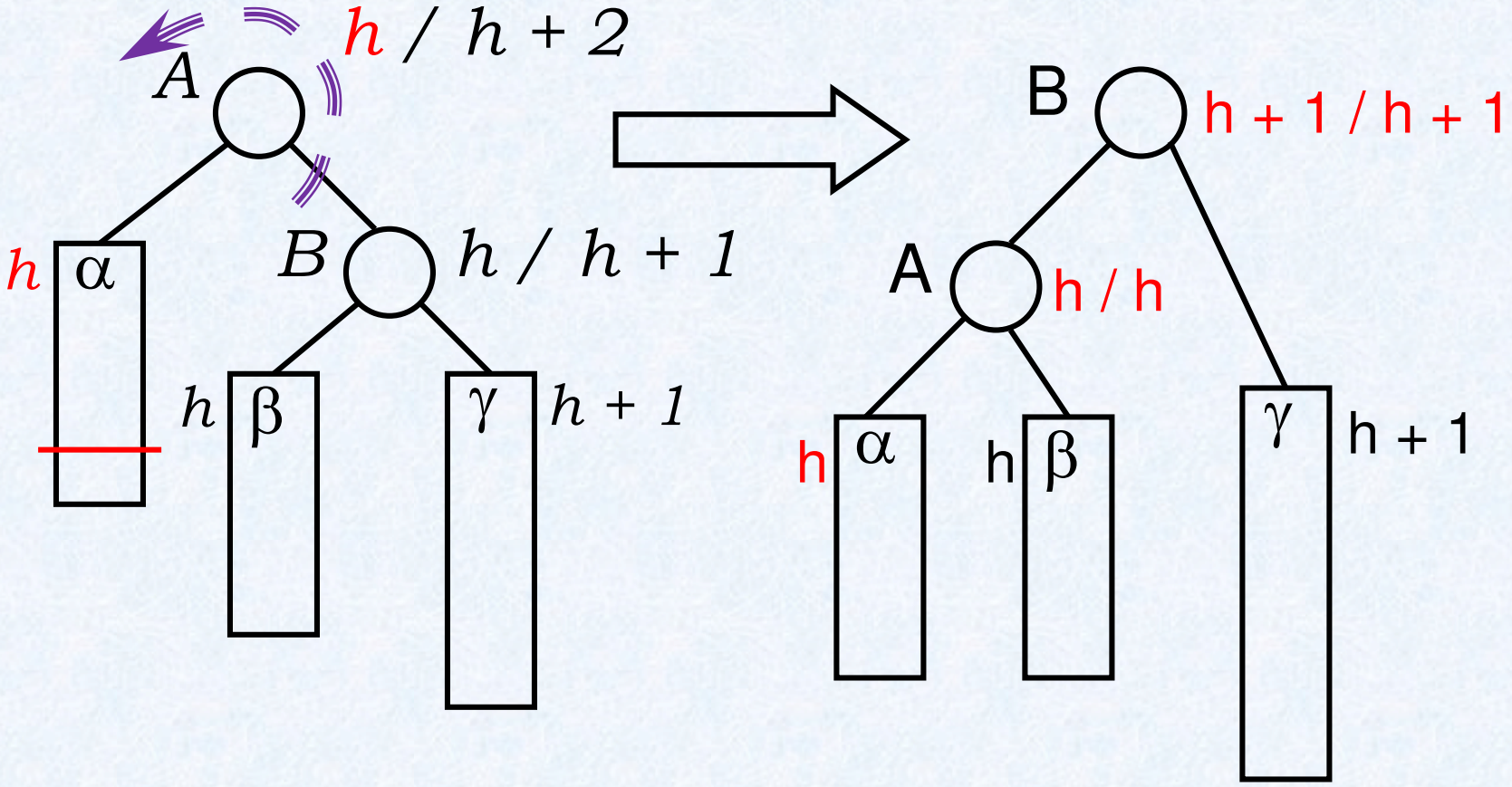
Удаление из поддерева α



Решение:

левый поворот вокруг узла A

Случай 1



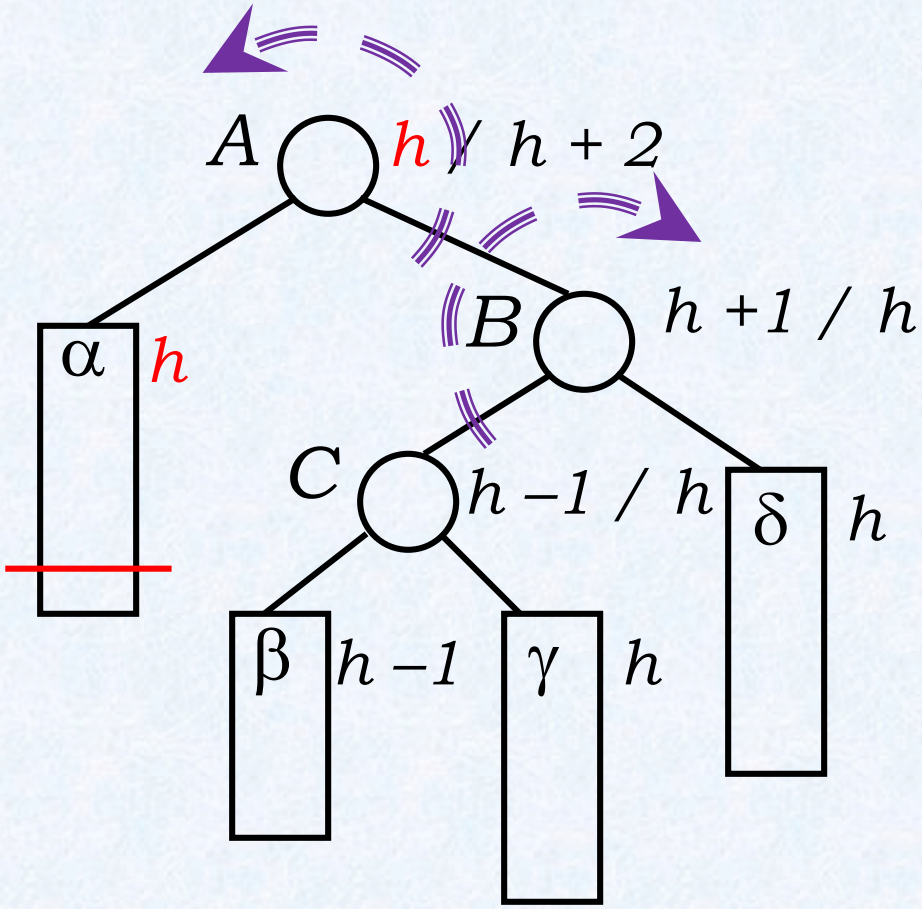
Случай 1

Высота поддерева уменьшилась !!!

Необходимо продолжить коррекцию узлов дерева, переустановив значение x

Случай 2а

Удаление из поддерева α



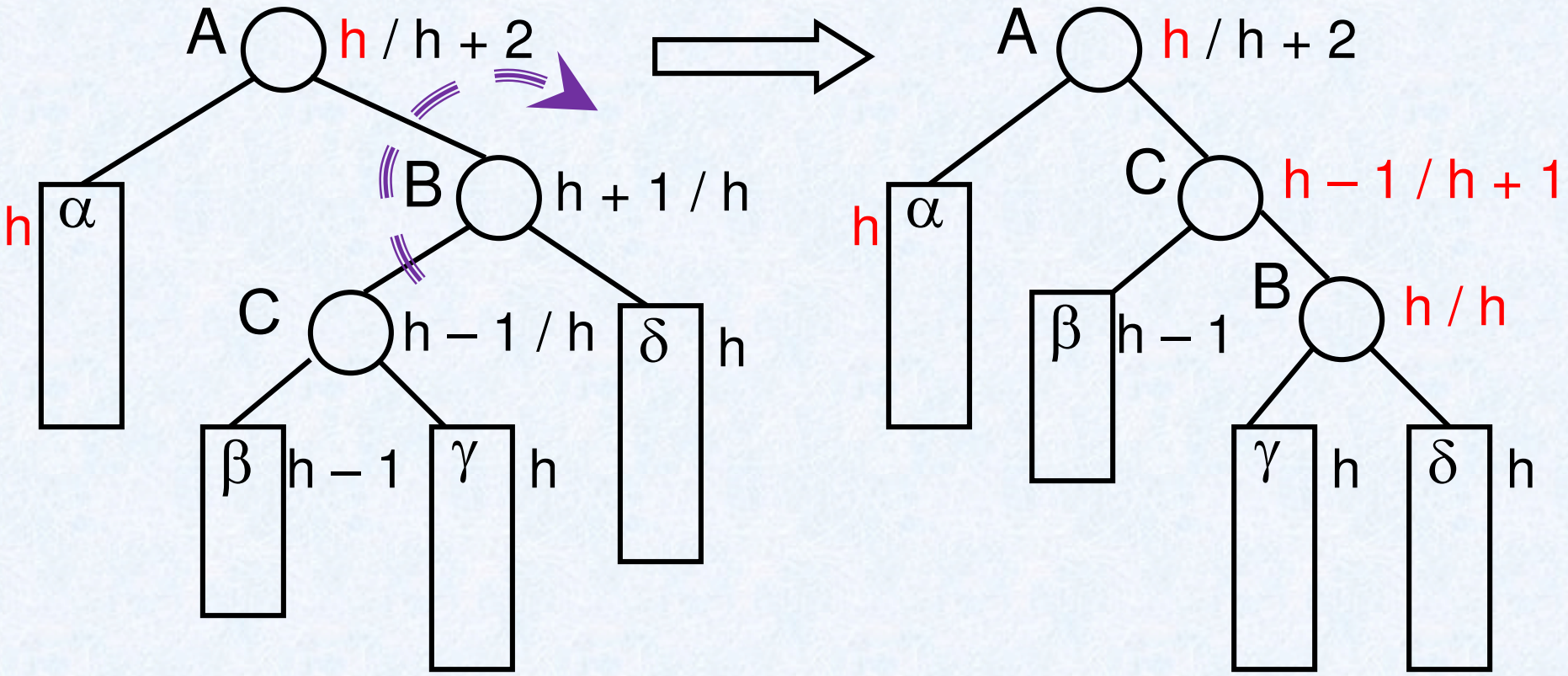
Решение:

правый поворот вокруг узла B

левый поворот вокруг узла A

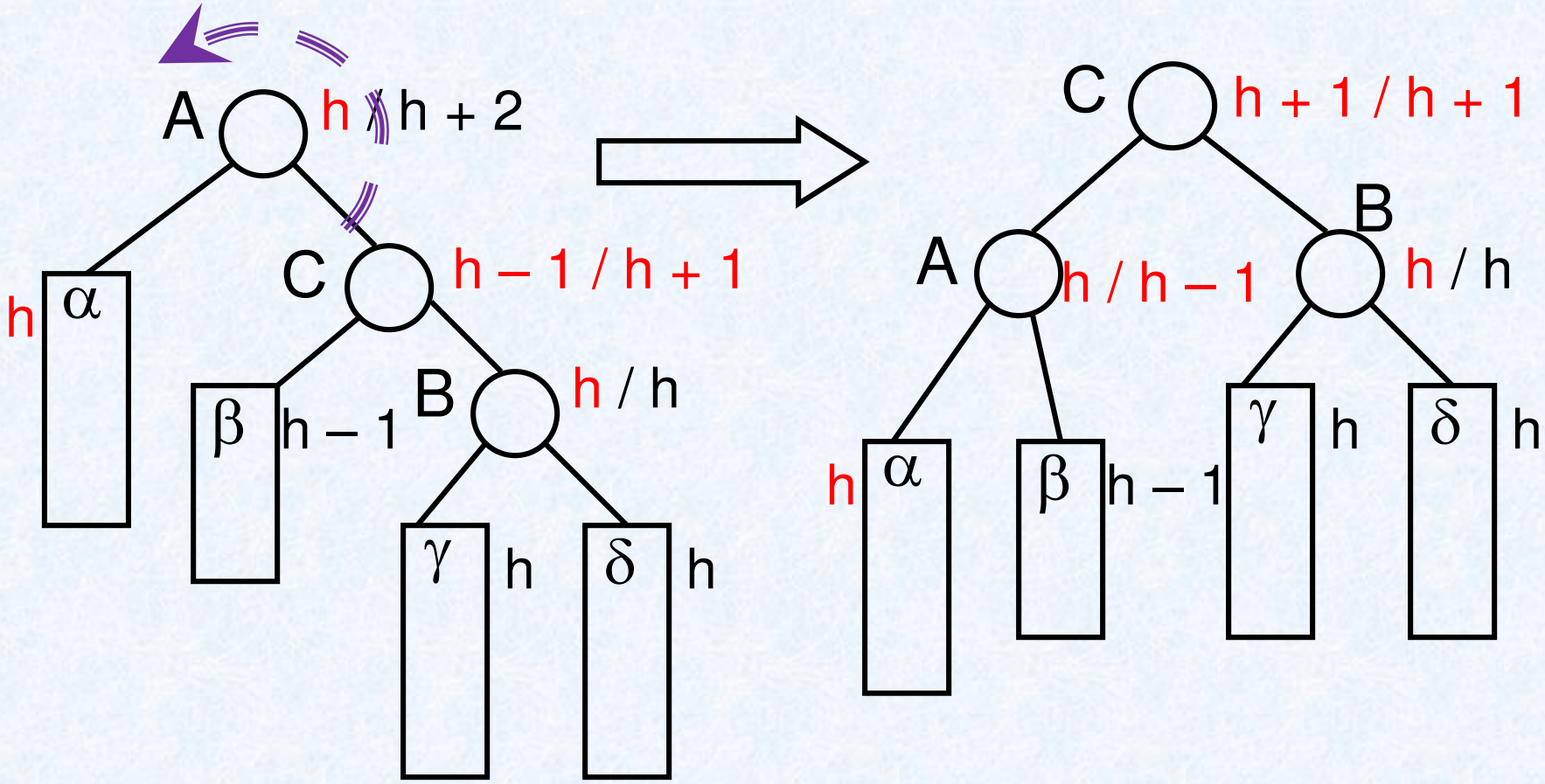
Случай 2а

Шаг 1



Случай 2а

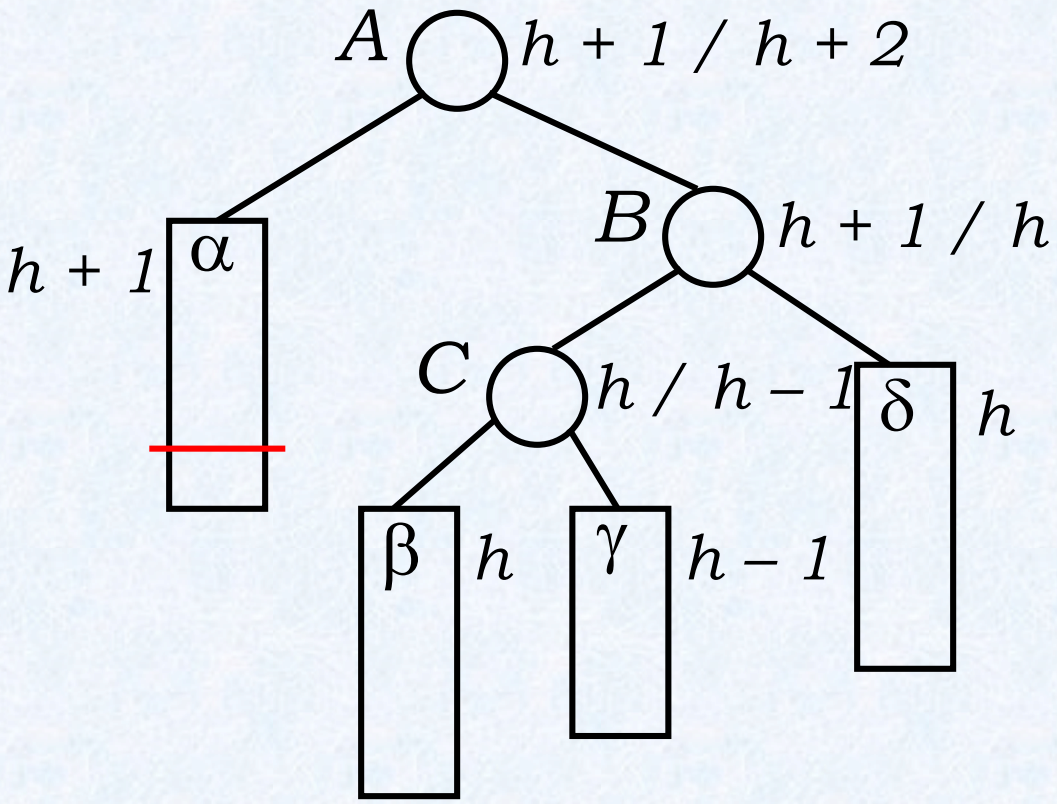
Шаг 2



Высота дерева уменьшилась на 1 !!!

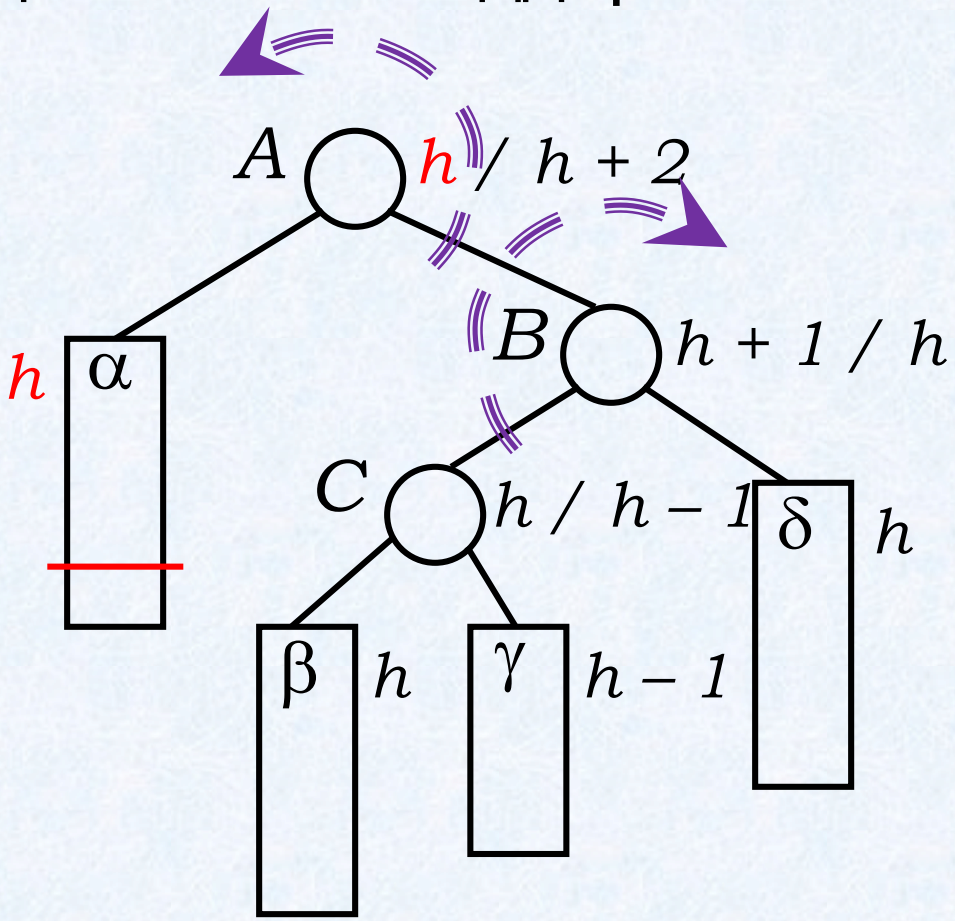
Случай 2б

Удаление из поддерева α



Случай 2б

Удаление из поддерева α



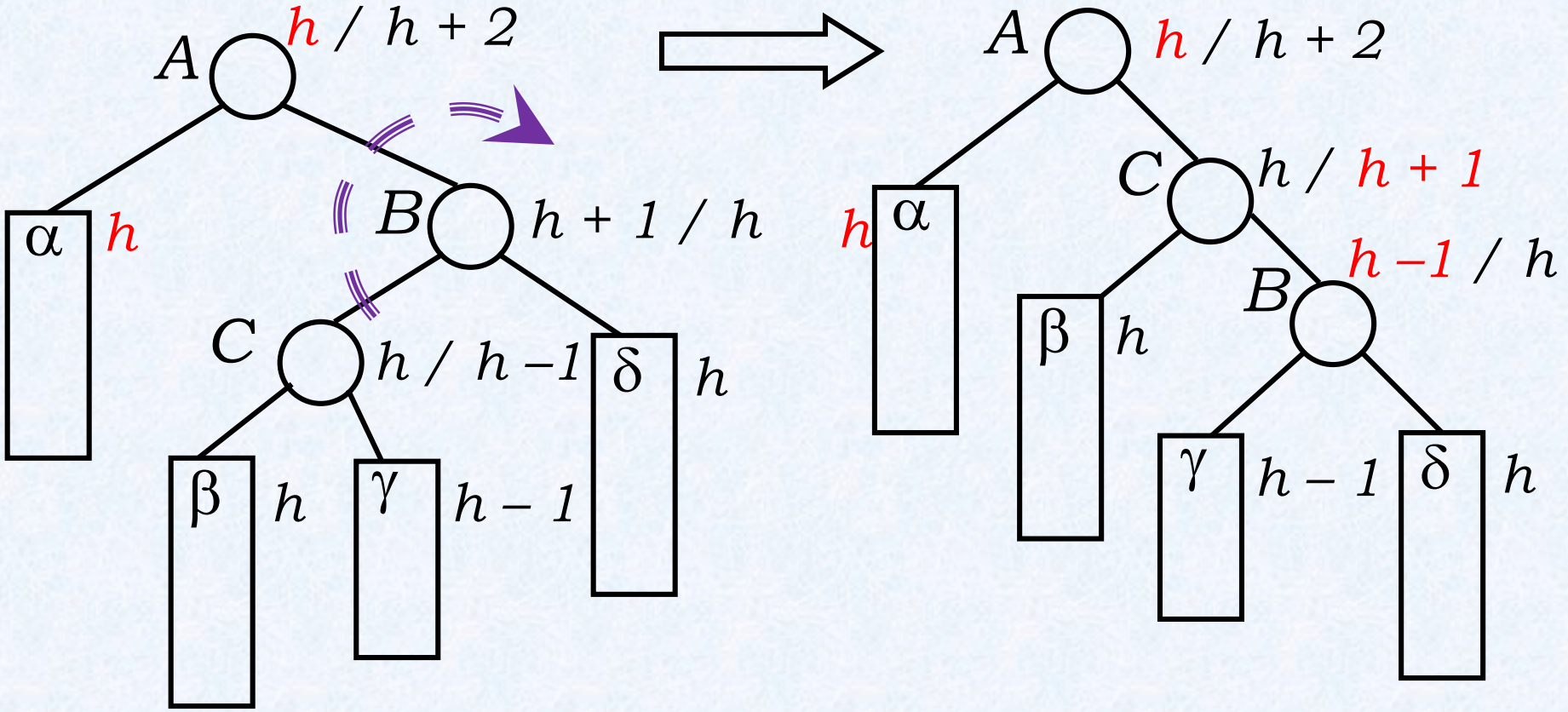
Решение:

правый поворот вокруг узла B

левый поворот вокруг узла A

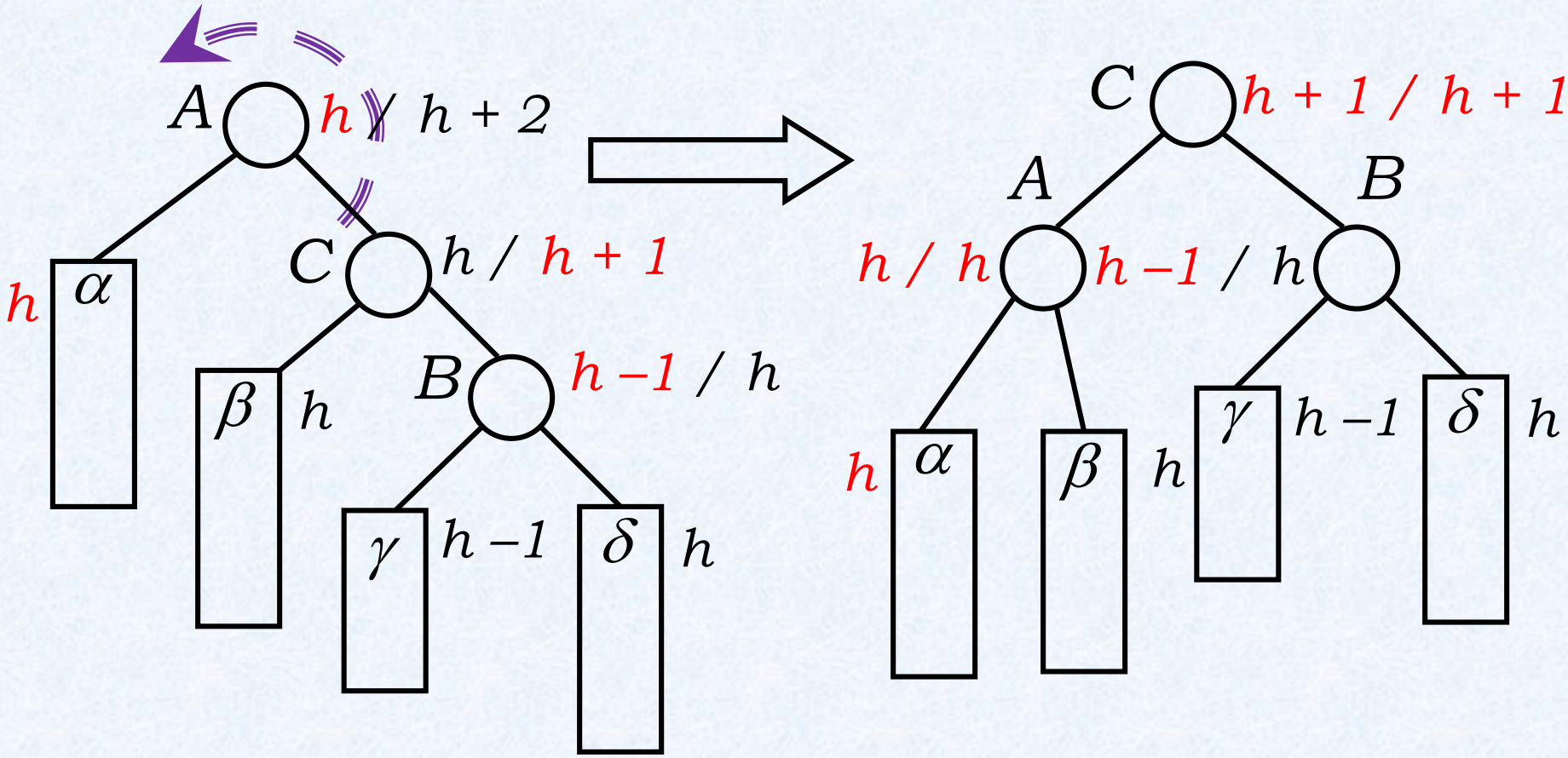
Случай 2б

Шаг 1



Случай 2б

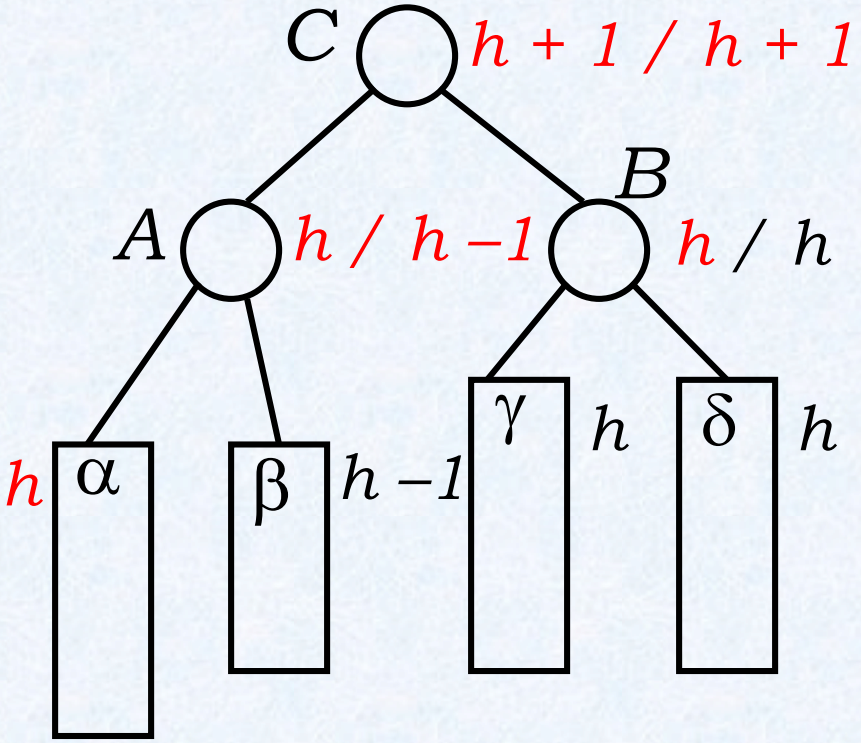
Шаг 2



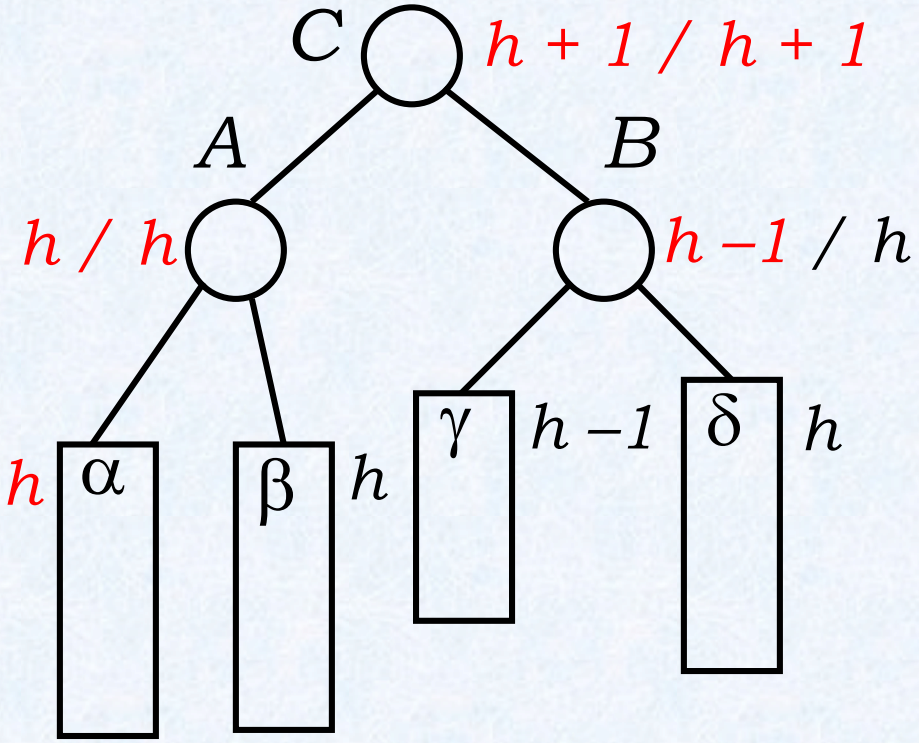
Высота дерева уменьшилась на 1 !!!

Случаи 2а и 2б

Случай 2а



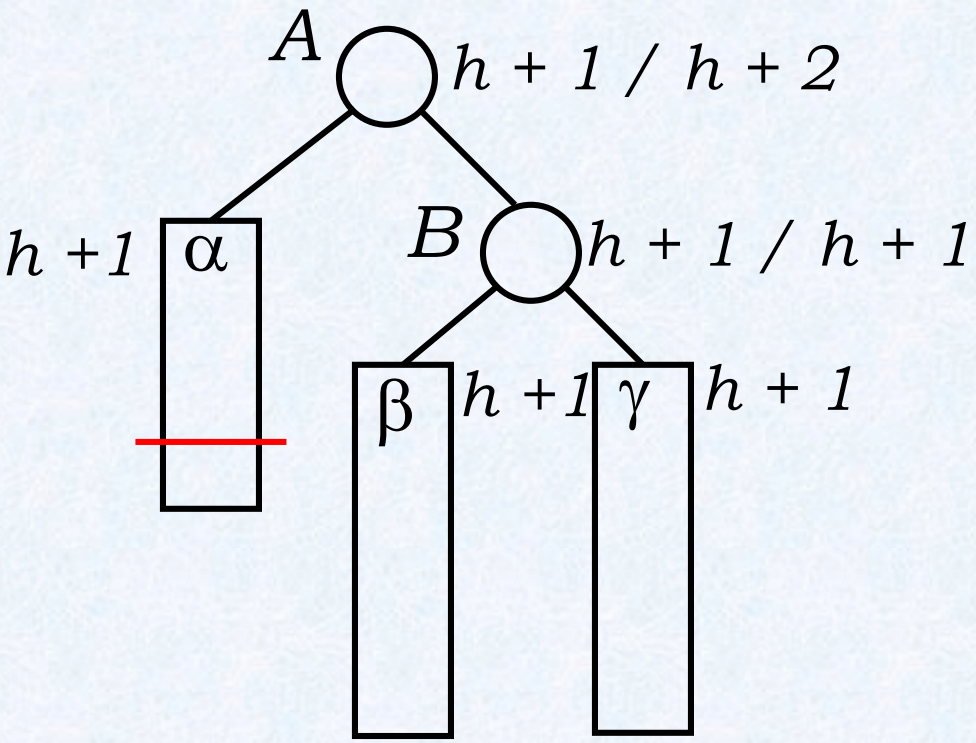
Случай 2б



Высота дерева уменьшилась на 1 !!!

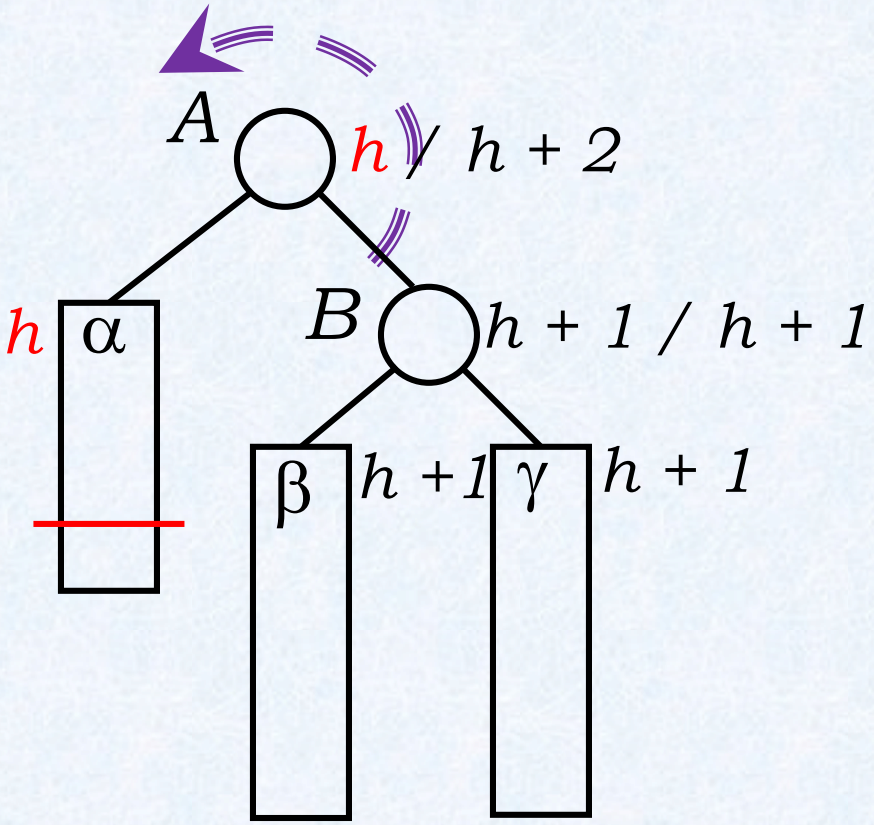
Случай 3

Удаление из поддерева α



Случай 3

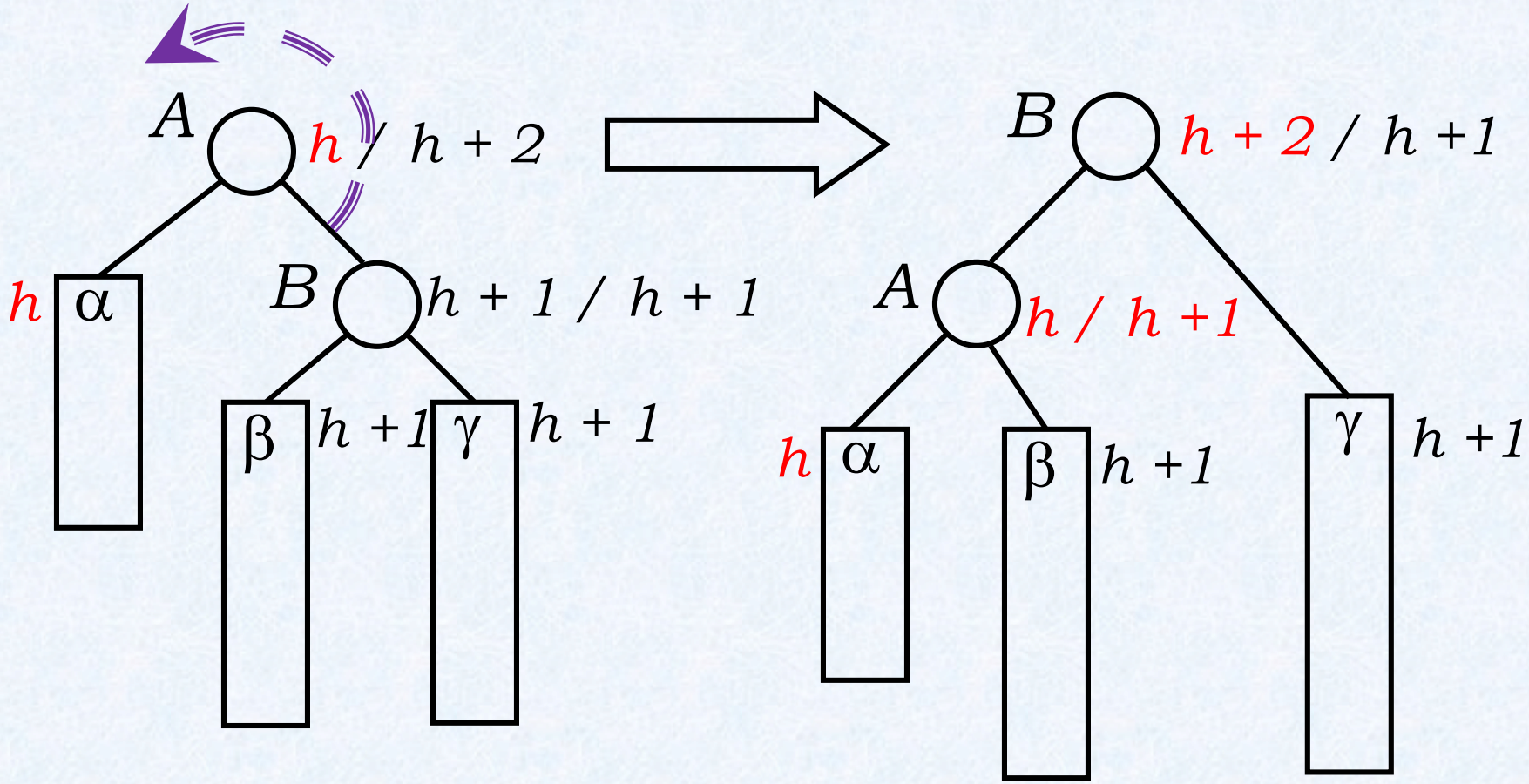
Удаление из поддерева α



Решение:

левый поворот вокруг узла A

Случай 3



Высота дерева не изменилась