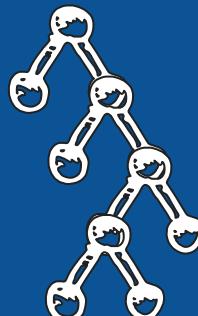


# Алгоритмы и Алгоритмические Языки

Семинар #16.2:

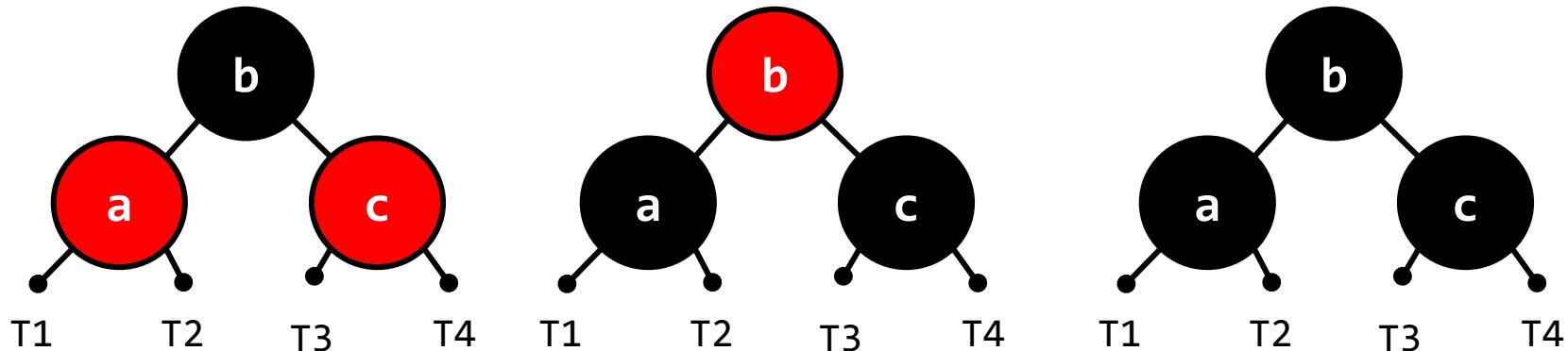
1. Свойства и структура красно-чёрного дерева.
2. Балансировка красно-чёрного дерева.
3. Задачи к экзамену.

# Свойства и структура красно-чёрного дерева



# Свойства красно-чёрного дерева

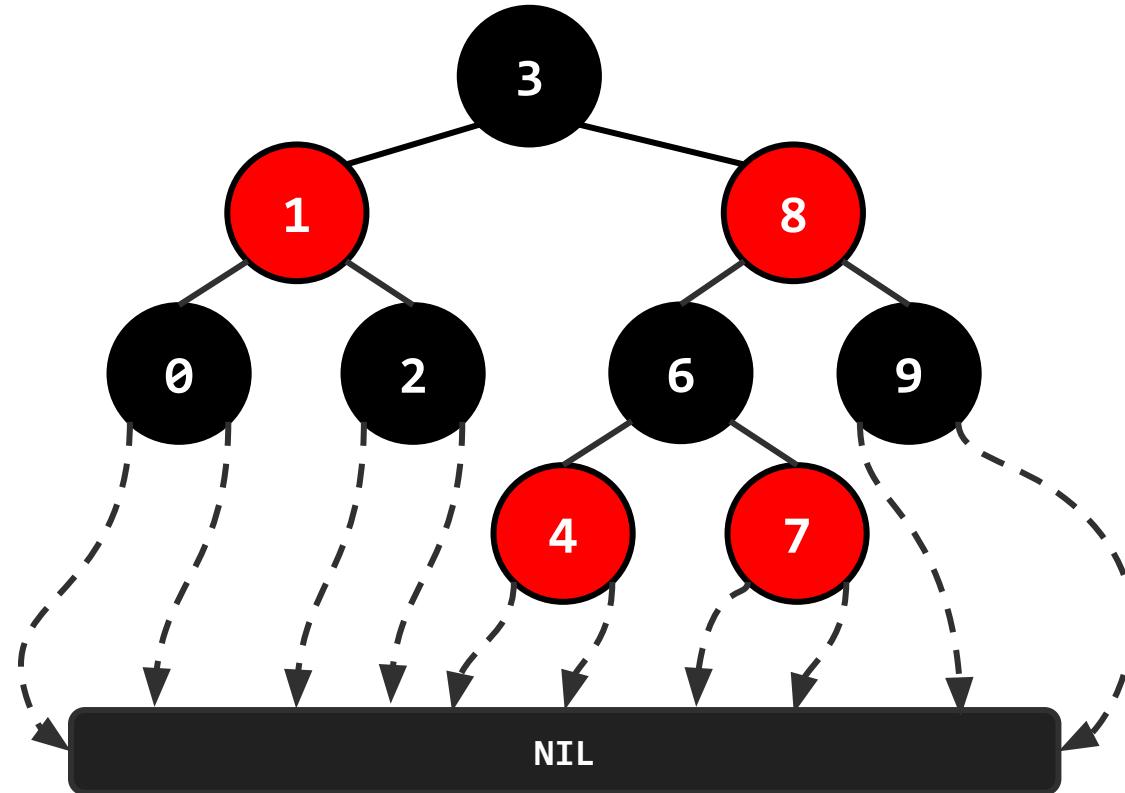
1. Каждый узел дерева либо **ЧЁРНЫЙ**, либо **КРАСНЫЙ**.
2. Корневой узел дерева всегда **ЧЁРНЫЙ**.
3. Каждый листовой узел (NIL) – **ЧЁРНЫЙ**.
4. Если узел **КРАСНЫЙ**, то оба его дочерних узла **ЧЁРНЫЕ**.



5. Для каждого узла: все пути к листьям содержат одинаковое кол-во **ЧЁРНЫХ** узлов.

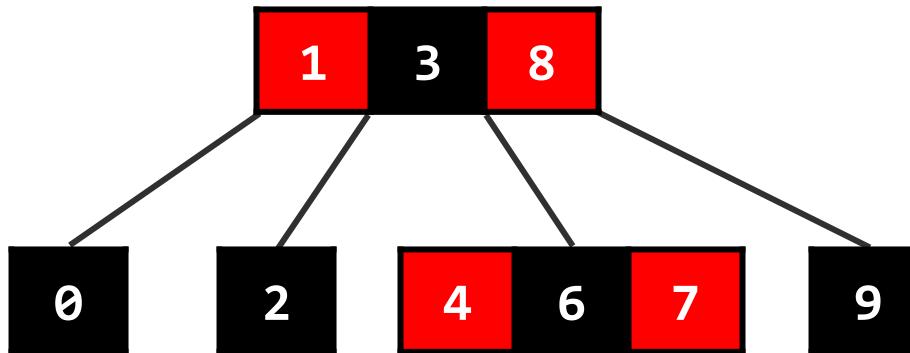
# Пример красно-чёрного дерева

Вставим ключи 0, 9, 1, 8, 2, 7, 3, 6, 4 в пустое дерево:



# Изоморфизм с В-деревьями

Есть соответствие между 2-3-4-деревьями и КЧ-деревьями:



## Правило №4

**КРАСНЫЕ** узлы присоединяются к родительским **ЧЁРНЫМ** узлам.  
Получаются узлы с 2, 3, 4 родительскими узлами.

## Правило №5

У В-дерева все листовые узлы расположены на одном уровне.

# Высота красно-чёрного дерева

$bh(x) = \text{black-height}(x)$  – количество чёрных узлов на путях от  $x$  до NIL (без учёта  $x$  и NIL).

Дерево, начинающееся с  $x$ , имеет хотя бы  $2^{bh(x)} - 1$  узлов.

Пусть дерево целиком имеет высоту  $h$ .

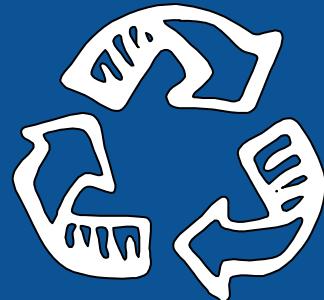
Т.к. не более половины узлов красные, то  $bh(\text{root}) \geq h/2$ .

Тогда:  $n \geq 2^{bh(\text{root})} - 1 \geq 2^{h/2} - 1$ .

Тогда:  $h \leq 2 \cdot \log_2(n + 1)$ .

Операции вставки и удаления эффективны!

# Балансировка красно- чёрного дерева



# Вставка в красно-чёрное дерево

```
RB-INSERT( $T, z$ )
1    $y = T.nil$ 
2    $x = T.root$ 
3   while  $x \neq T.nil$ 
4        $y = x$ 
5       if  $z.key < x.key$ 
6            $x = x.left$ 
7       else  $x = x.right$ 
8    $z.p = y$ 
9   if  $y == T.nil$ 
10       $T.root = z$ 
11   elseif  $z.key < y.key$ 
12       $y.left = z$ 
13   else  $y.right = z$ 
14    $z.left = T.nil$ 
15    $z.right = T.nil$ 
16    $z.color = \text{RED}$ 
17   RB-INSERT-FIXUP( $T, z$ )
```

ToDo: корректность

# Вставка в красно-чёрное дерево

```
RB-INSERT-FIXUP( $T, z$ )
```

```
1  while  $z.p.color == \text{RED}$ 
2      if  $z.p == z.p.p.left$ 
3           $y = z.p.p.right$ 
4          if  $y.color == \text{RED}$ 
5               $z.p.color = \text{BLACK}$ 
6               $y.color = \text{BLACK}$ 
7               $z.p.p.color = \text{RED}$ 
8               $z = z.p.p$ 
9      else if  $z == z.p.right$ 
10          $z = z.p$ 
11         LEFT-ROTATE( $T, z$ )
12          $z.p.color = \text{BLACK}$ 
13          $z.p.p.color = \text{RED}$ 
14         RIGHT-ROTATE( $T, z.p.p$ )
15     else (same as then clause
           with “right” and “left” exchanged)
16      $T.root.color = \text{BLACK}$ 
```

ToDo: корректность

# Удаление из красно-чёрного дерева

RB-DELETE( $T, z$ )

```
1   $y = z$ 
2   $y\text{-original-color} = y.\text{color}$ 
3  if  $z.\text{left} == T.\text{nil}$ 
4       $x = z.\text{right}$ 
5      RB-TRANSPLANT( $T, z, z.\text{right}$ )
6  elseif  $z.\text{right} == T.\text{nil}$ 
7       $x = z.\text{left}$ 
8      RB-TRANSPLANT( $T, z, z.\text{left}$ )
9  else  $y = \text{TREE-MINIMUM}(z.\text{right})$ 
10      $y\text{-original-color} = y.\text{color}$ 
11      $x = y.\text{right}$ 
12     if  $y.p == z$ 
13          $x.p = y$ 
14     else RB-TRANSPLANT( $T, y, y.\text{right}$ )
15          $y.\text{right} = z.\text{right}$ 
16          $y.\text{right}.p = y$ 
17     RB-TRANSPLANT( $T, z, y$ )
18      $y.\text{left} = z.\text{left}$ 
19      $y.\text{left}.p = y$ 
20      $y.\text{color} = z.\text{color}$ 
21 if  $y\text{-original-color} == \text{BLACK}$ 
22     RB-DELETE-FIXUP( $T, x$ )
```

ToDo: корректность

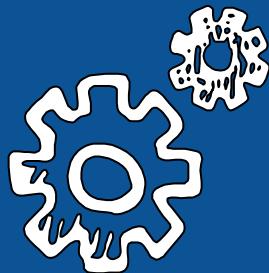
# Удаление из красно-чёрного дерева

```
RB-DELETE-FIXUP( $T, x$ )
```

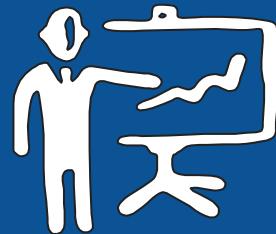
```
1  while  $x \neq T.root$  and  $x.color == \text{BLACK}$ 
2    if  $x == x.p.left$ 
3       $w = x.p.right$ 
4      if  $w.color == \text{RED}$ 
5         $w.color = \text{BLACK}$ 
6         $x.p.color = \text{RED}$ 
7        LEFT-ROTATE( $T, x.p$ )
8         $w = x.p.right$ 
9      if  $w.left.color == \text{BLACK}$  and  $w.right.color == \text{BLACK}$ 
10         $w.color = \text{RED}$ 
11         $x = x.p$ 
12      else if  $w.right.color == \text{BLACK}$ 
13         $w.left.color = \text{BLACK}$ 
14         $w.color = \text{RED}$ 
15        RIGHT-ROTATE( $T, w$ )
16         $w = x.p.right$ 
17         $w.color = x.p.color$ 
18         $x.p.color = \text{BLACK}$ 
19         $w.right.color = \text{BLACK}$ 
20        LEFT-ROTATE( $T, x.p$ )
21         $x = T.root$ 
22    else (same as then clause with “right” and “left” exchanged)
23     $x.color = \text{BLACK}$ 
```

ToDo: корректность

# Задачи к экзамену



# Вопросы?



Красивые иконки взяты с сайта [handdrawngoods.com](http://handdrawngoods.com)