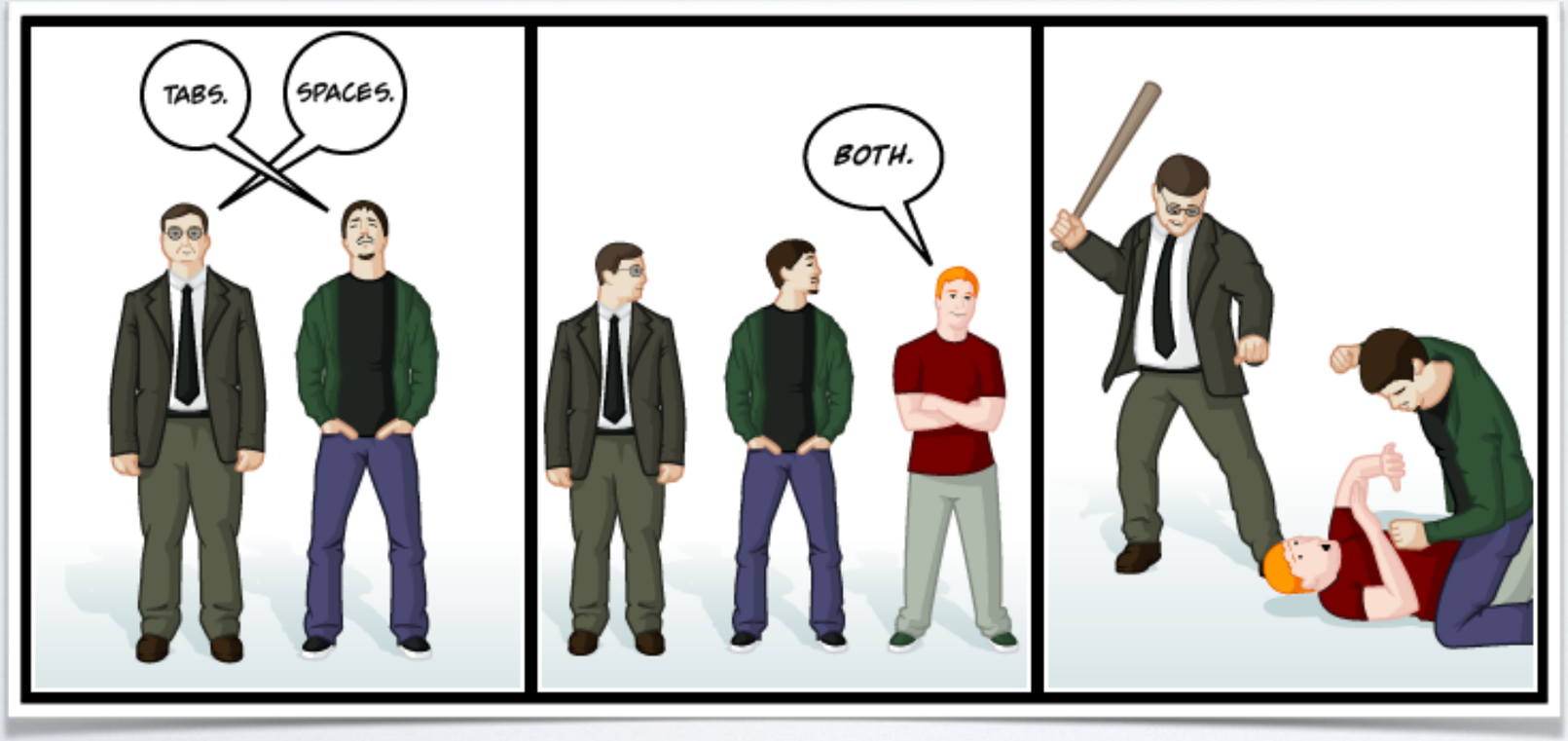
ОСНОВЫ ПРОГРАММНОГО КОНСТРУИРОВАНИЯ

Лекция № 7

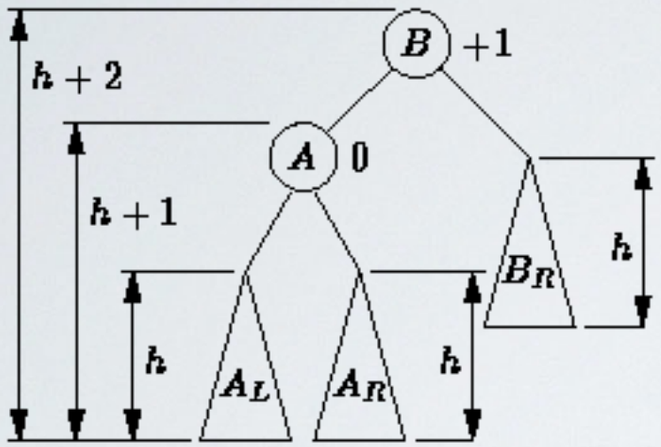
23 марта 2020 г.



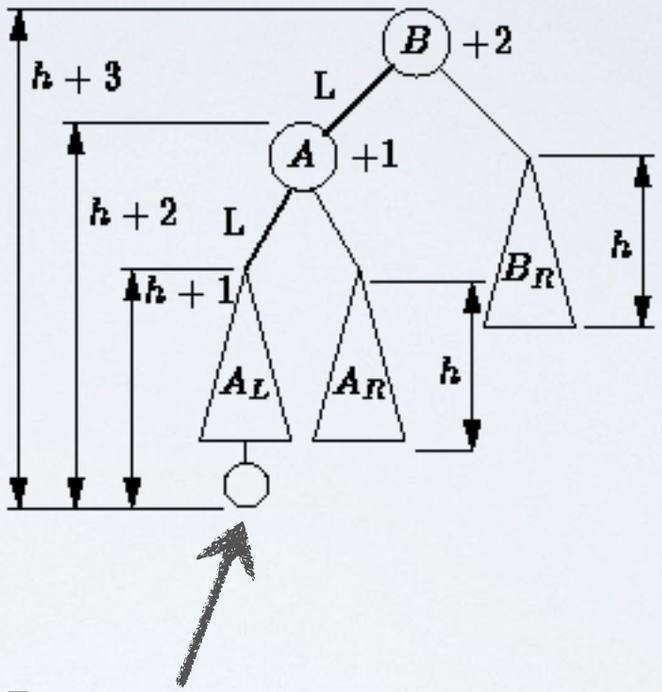
АВЛ-ДЕРЕВЬЯ

- 1962 г. Адельсон-Вельский и Ландис (СССР)
- **Сбалансированное дерево:** высоты двух родственных поддеревьев отличаются не более, чем на единицу
- **Перебалансировка** после операций вставки и удаления, нарушающих свойство сбалансированности. Идем снизу вверх (к корню), восстанавливая баланс.
- В узел добавляется показатель сбалансированности, равный разности высот поддеревьев (0, +1, −1).

БАЛАНСИРОВКА: МАЛЫЙ ЛЕВЫЙ ПОВОРОТ

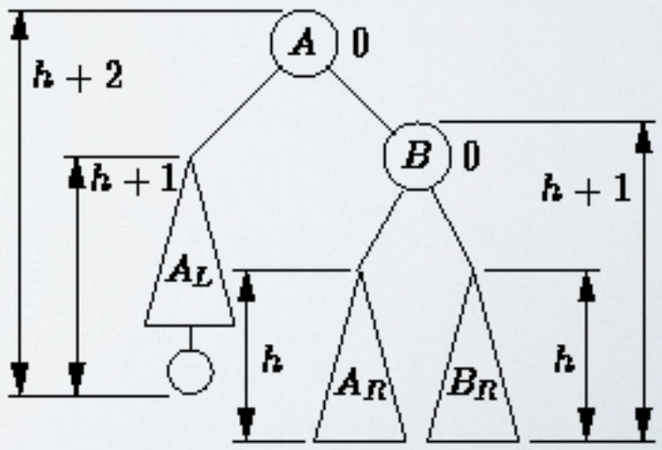


Было



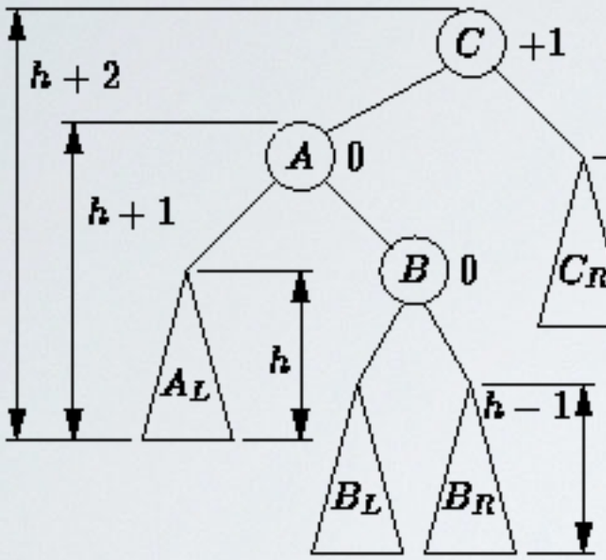
Вставили

Поворот вокруг B

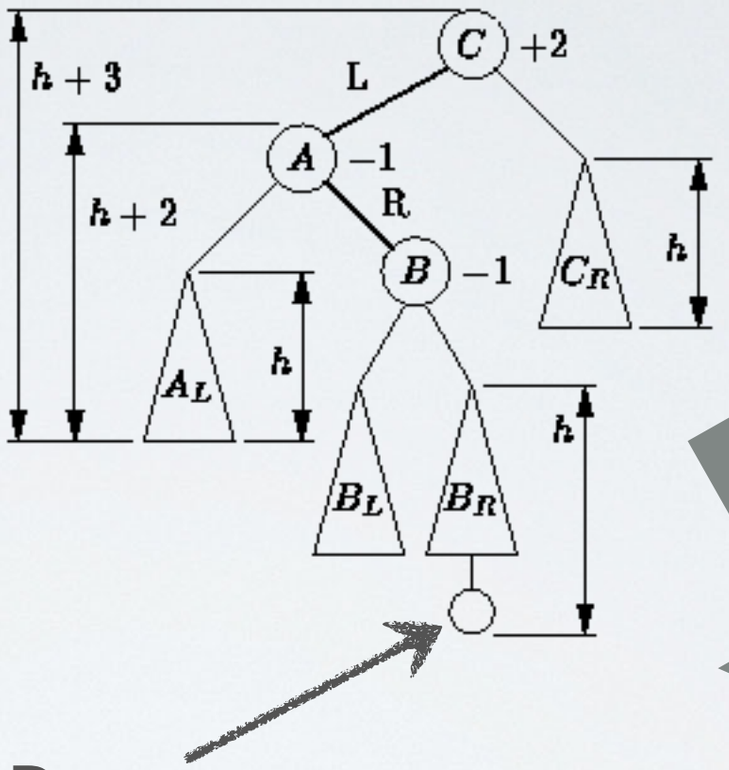


Выполняется, если B имеет баланс +2,
а A имеет баланс ≥ 0.

БАЛАНСИРОВКА: БОЛЬШОЙ ЛЕВЫЙ ПОВОРОТ

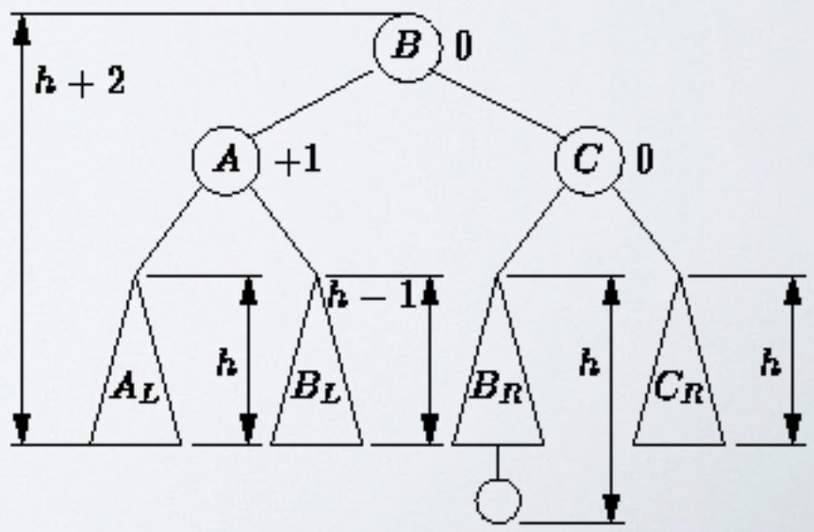


Было



Вставили

Поворот вокруг A, C

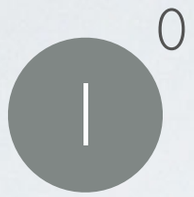


Выполняется, если C имеет баланс +2,
а A имеет баланс –1.

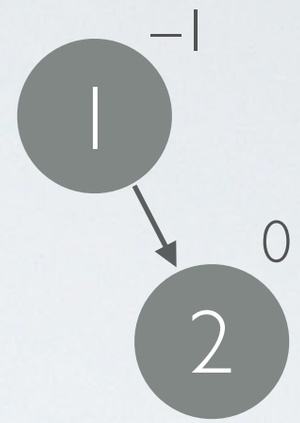
БАЛАНСИРОВКА: ПРАВЫЕ ПОВОРОТЫ

- Малый правый поворот аналогично малому левому
- Большой правый поворот аналогично большому левому

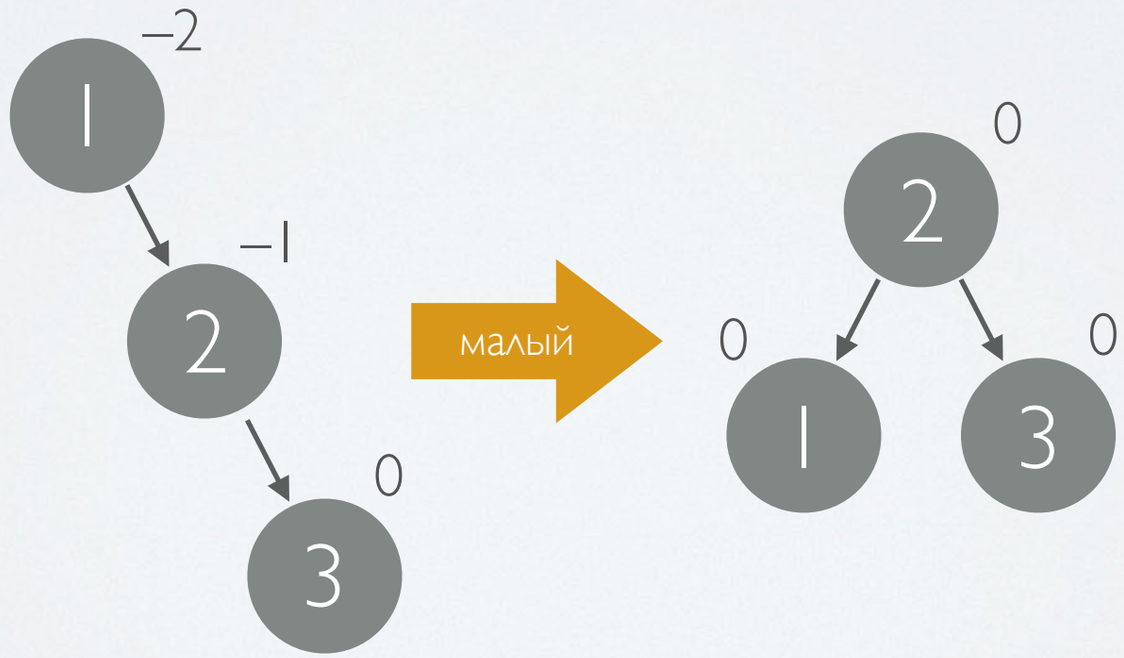
ПРИМЕР АВЛ-ДЕРЕВА



Вставляем 1

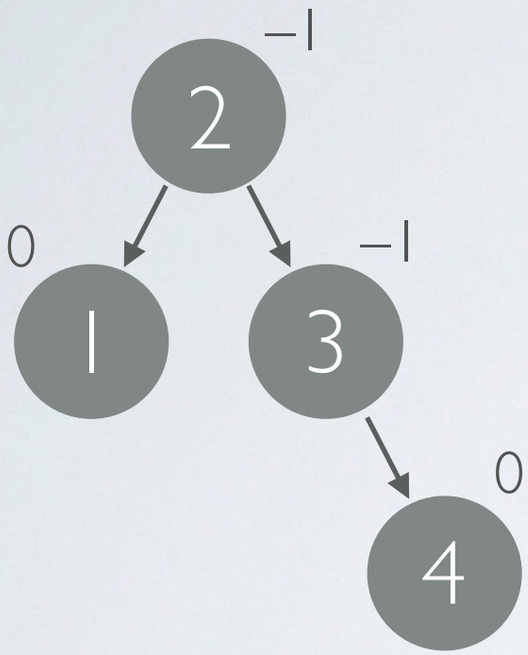


Вставляем 2

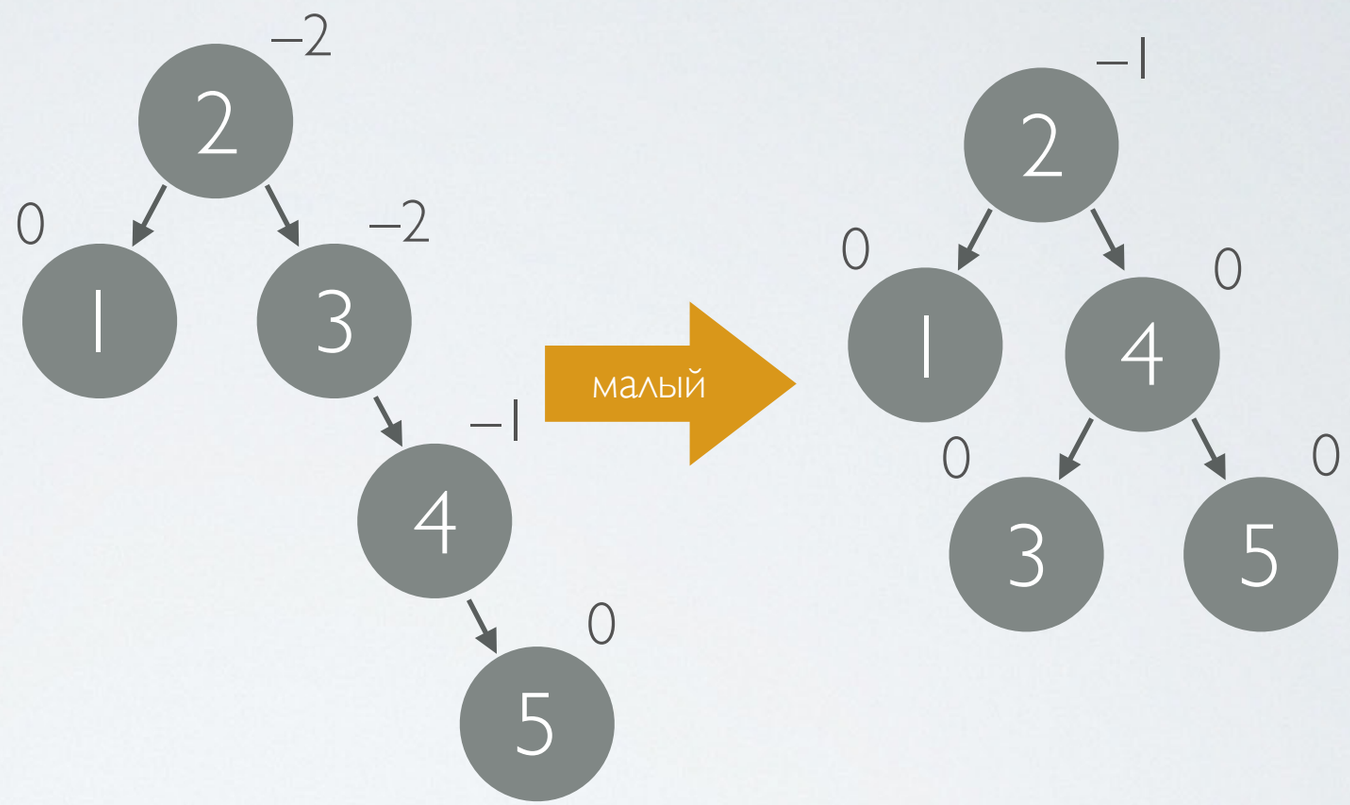


Вставляем 3

ПРИМЕР АВЛ-ДЕРЕВА

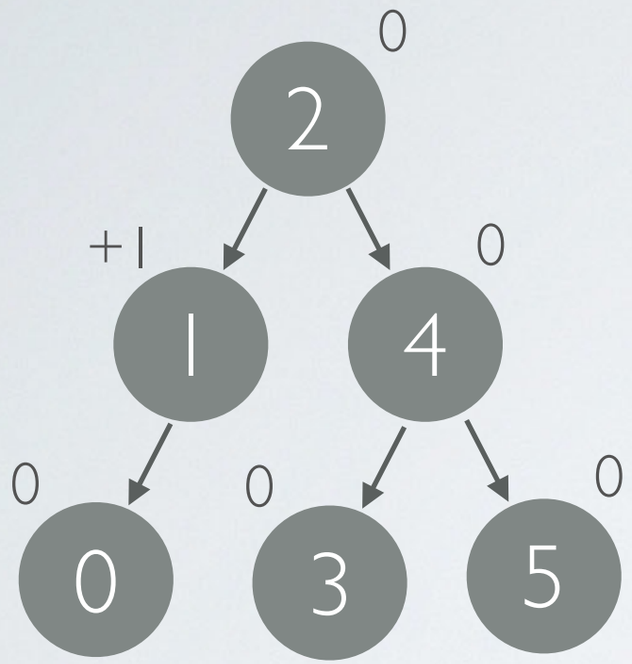


Вставляем 4



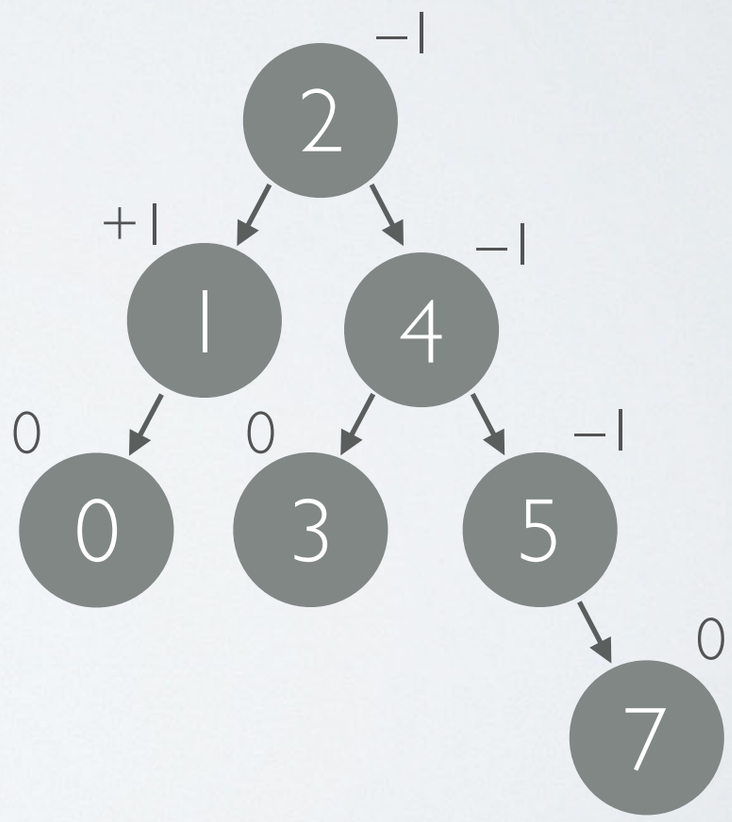
Вставляем 5

ПРИМЕР АВЛ-ДЕРЕВА

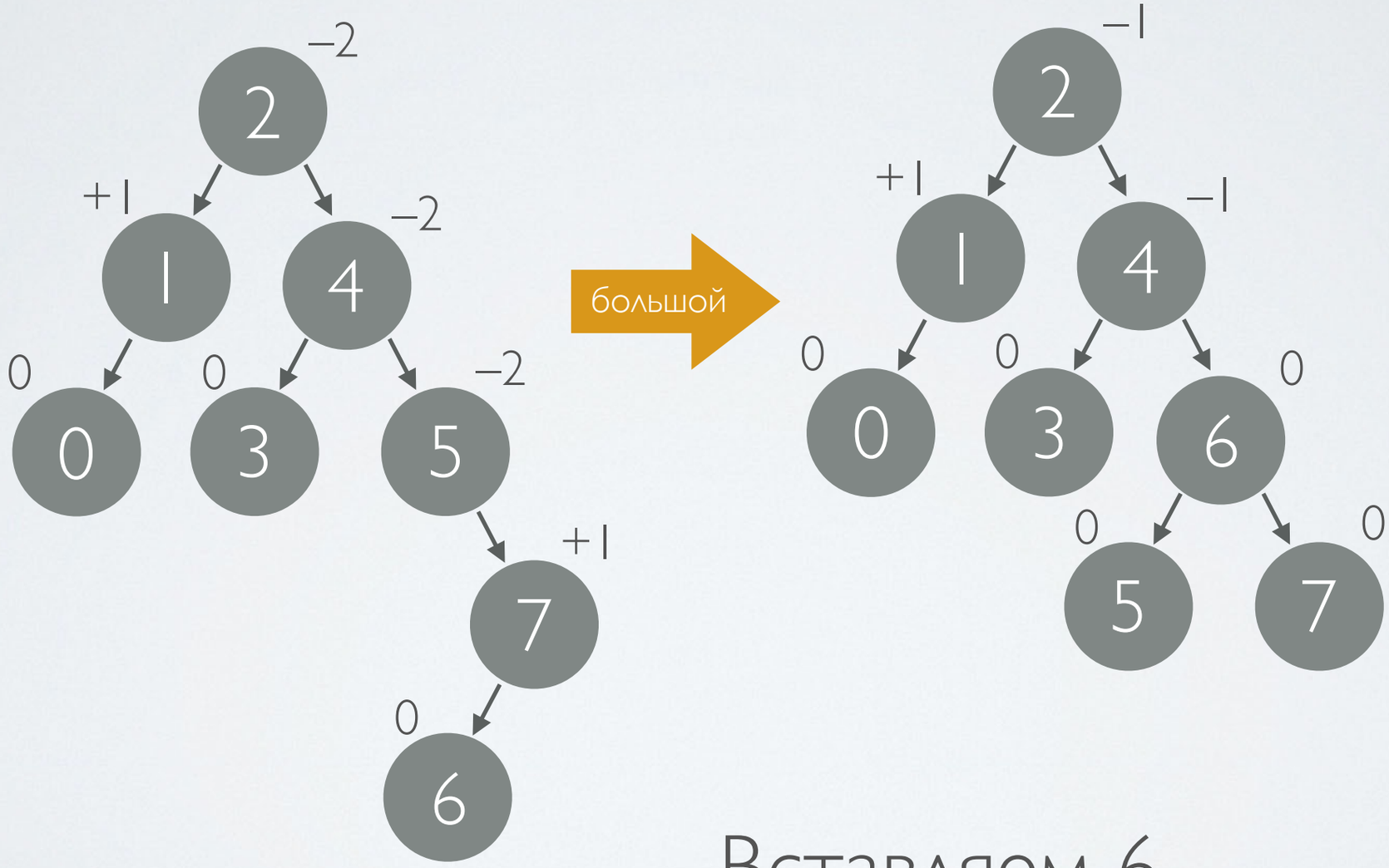


Вставляем 0

Вставляем 7

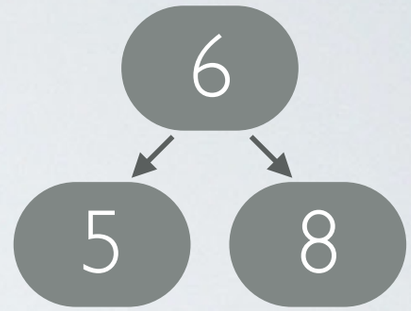


ПРИМЕР АВЛ-ДЕРЕВА

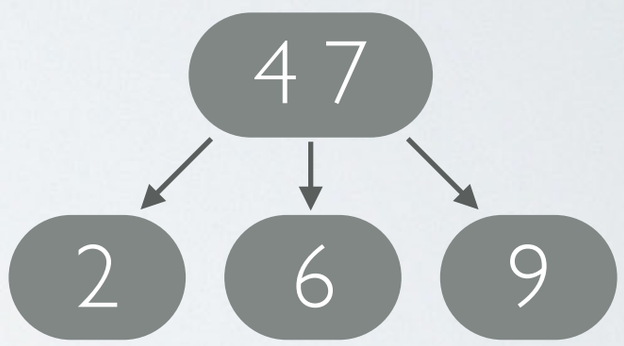


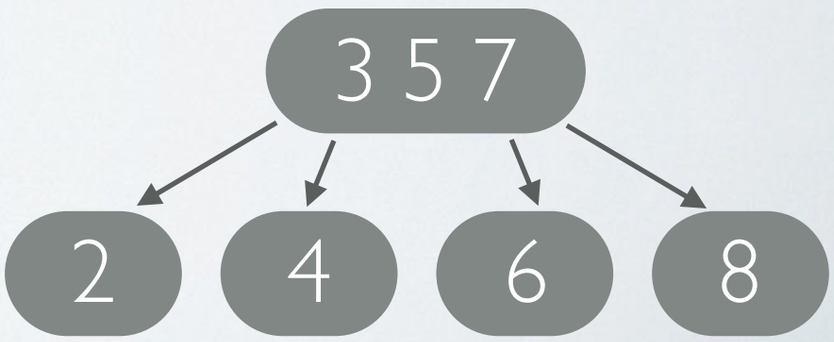
Вставляем 6

2-3-4 ДЕРЕВЬЯ



- Дерево поиска, узлы которого:
  - либо пусты;
  - либо 2-узел: 1 значение, 2 поддерева;
  - либо 3-узел: 2 значения, 3 поддерева;
  - либо 4-узел: 3 значения, 4 поддерева.
- Всегда идеально сбалансировано: высоты всех поддеревьев равны.

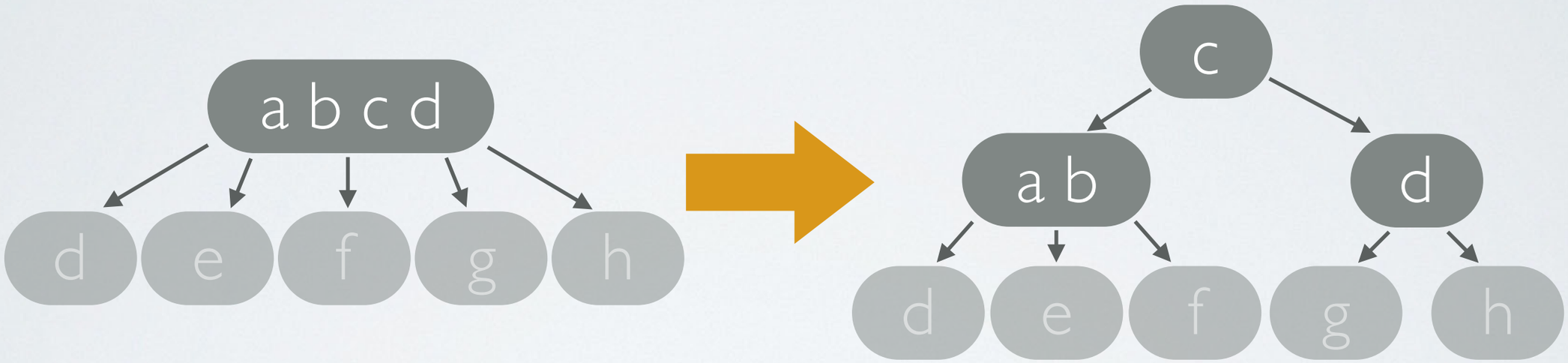




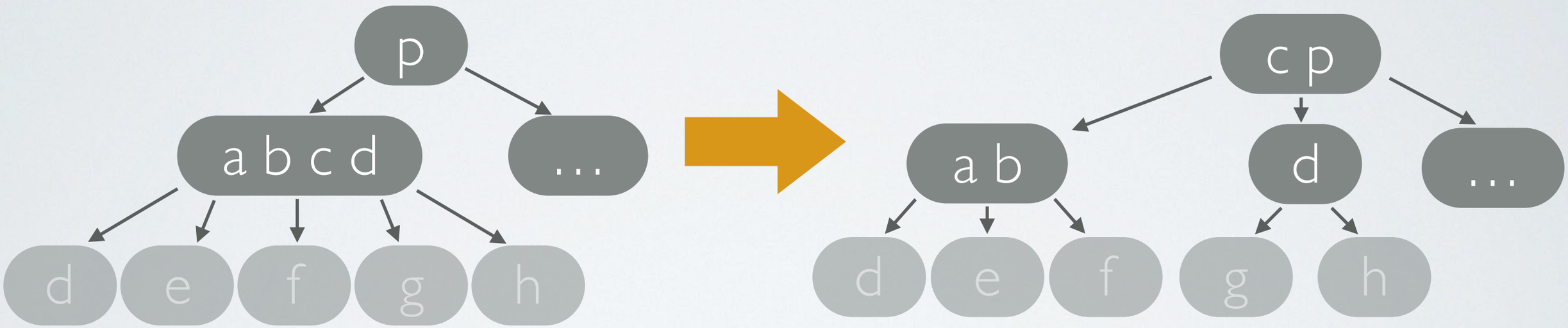
2-3-4 ДЕРЕВЬЯ: ПОИСК И ВСТАВКА

- Поиск как в обычном дереве поиска.
- Вставка в 2-узел: превращаем его в 3-узел.
- Вставка в 3-узел: превращаем его в 4-узел.
- Вставка в 4-узел: временно создаем 5-узел, вытаскиваем одно из значений и добавляем его в родителя.

ВСТАВКА:
5-УЗЕЛ КАК КОРЕНЬ



ВСТАВКА:
5-УЗЕЛ С РОДИТЕЛЕМ

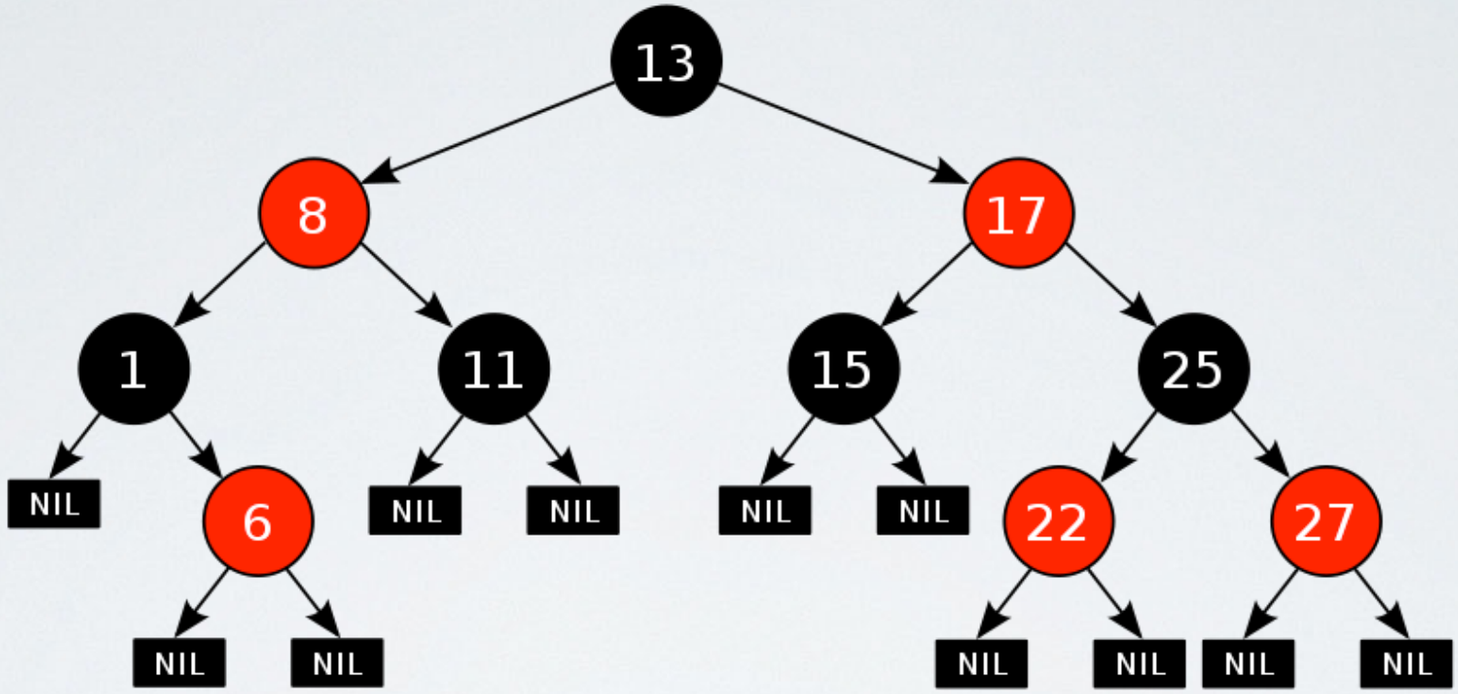


Вытягиваем одно из значений на уровень выше и продолжаем рекурсивно идти наверх, если получился новый 5-узел.

2-3-4 ДЕРЕВЬЯ: АНАЛИЗ

- Высота дерева: $\log_4(N) \leq h(N) \leq \log_2(N)$.
- Всегда идеально сбалансировано.
- Очень трудоемкая реализация, но идея-то хорошая!

КРАСНО-ЧЕРНЫЕ ДЕРЕВЬЯ
RED-BLACK TREES



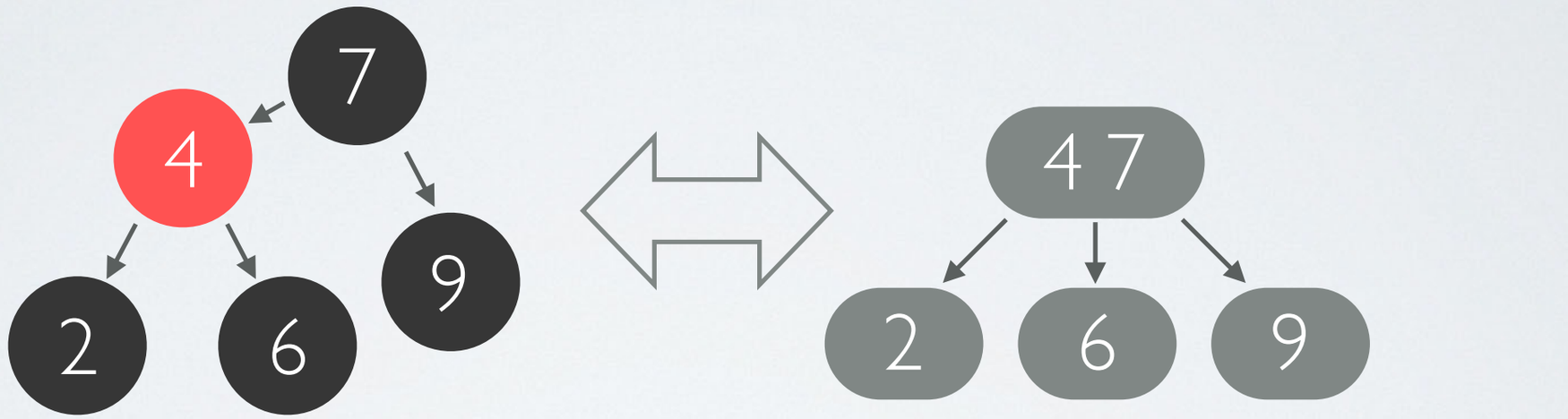
1. Все узлы либо **красные**, либо **черные**. Корень **черный**.
2. Потомки **красного** узла **черные**.
3. Все листья (NIL) **черные**.
4. Пути от любого узла до потомков содержат одинаковое количество **черных** узлов.
5. **(Следствие)** Пути от корня до двух любых узлов отличаются не более чем в 2 раза.

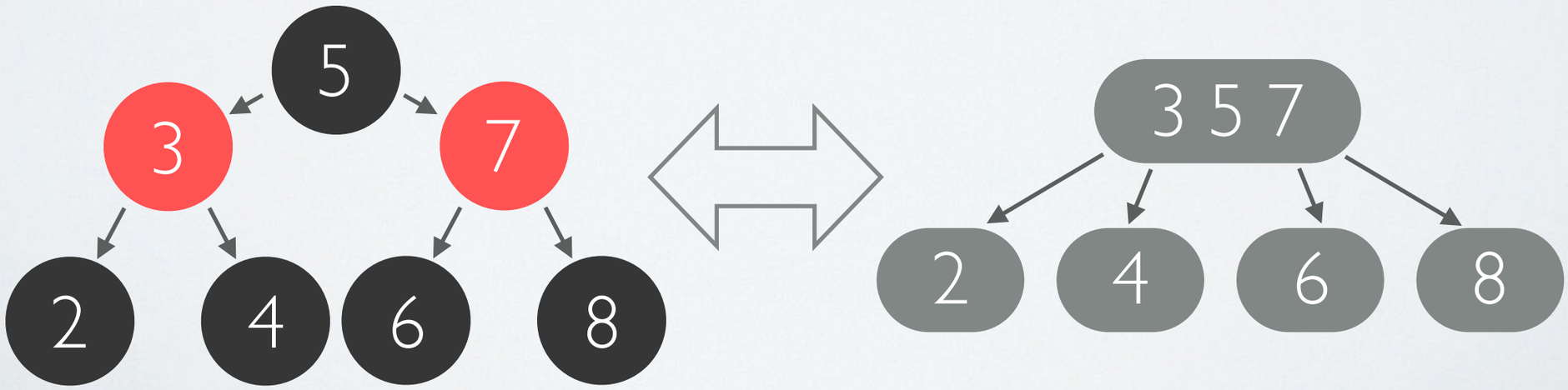
RB TREE: АНАЛИЗ

- Описаны и изучены в 1970-ые, с тех пор стандарт де-факто.
- Производительность сравнима с AVL-деревьями.
- Реализация сложна. Шесть возможных случаев вставки и симметричные им…
  - И еще столько же на удаление…

RB КАК 2-3-4

Красный узел будем интерпретировать как часть родителя, а не как отдельный узел:

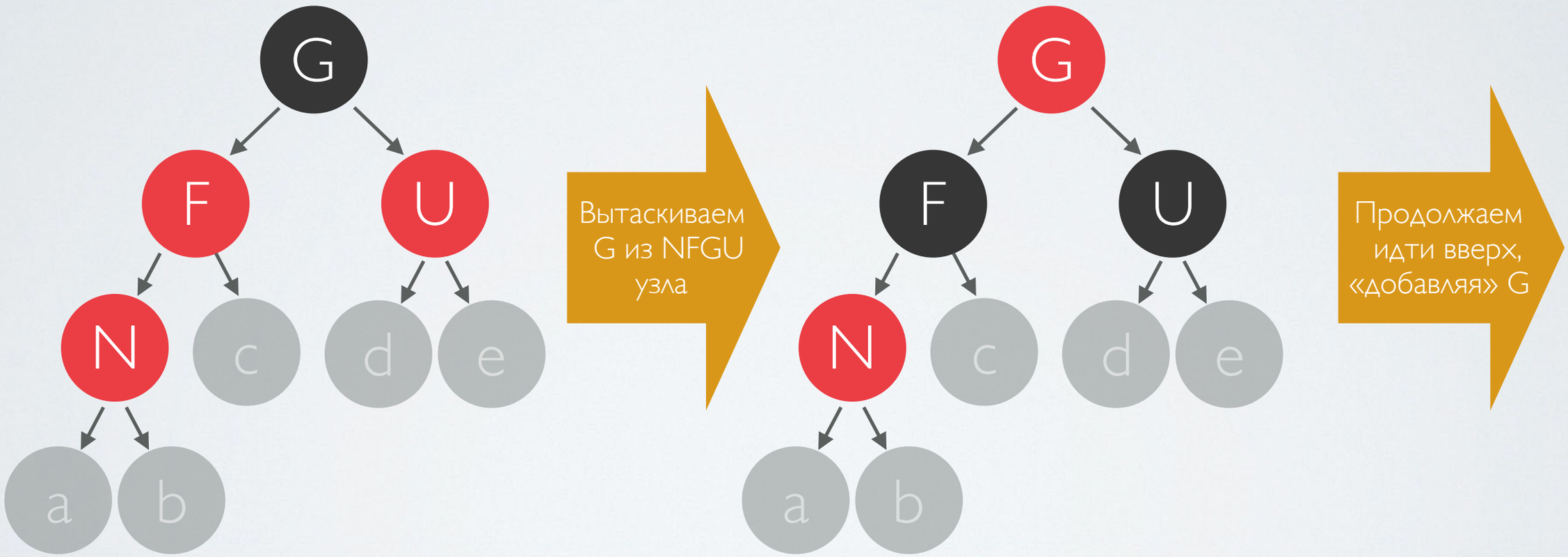




RB ВСТАВКА

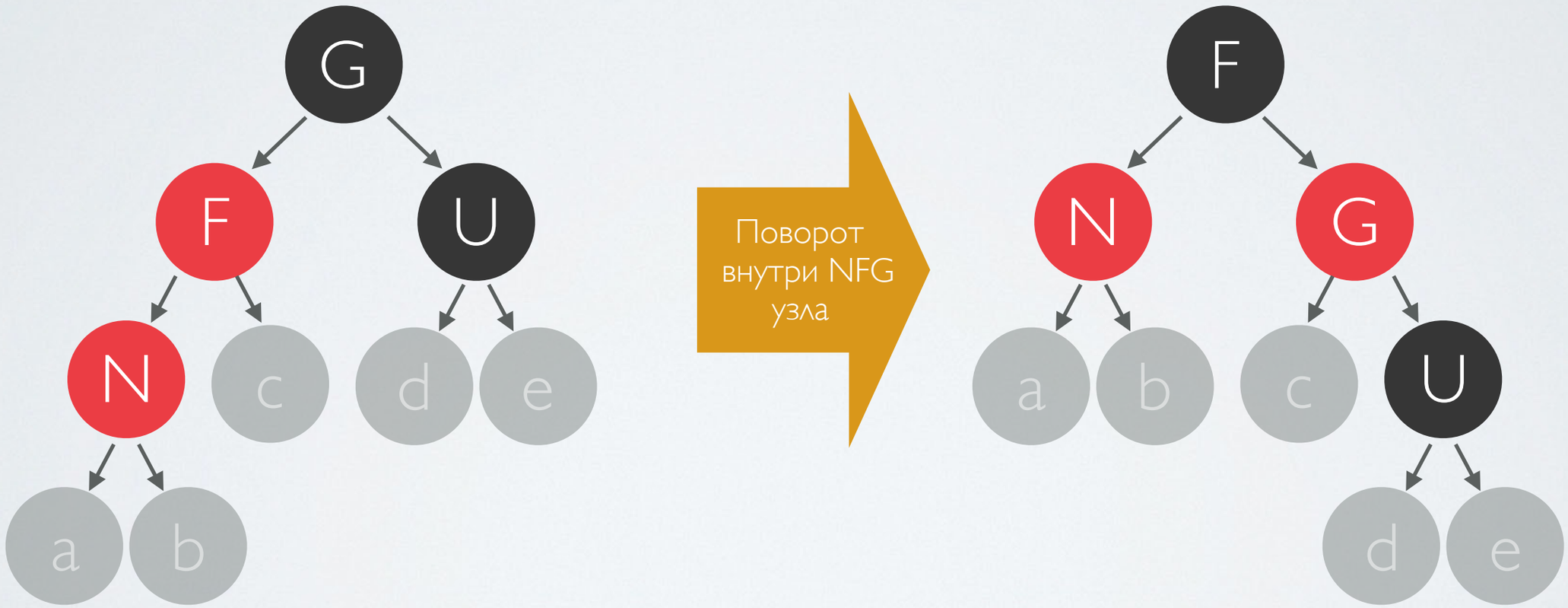
- Вставляемый узел – **красный**.
- Вставка в корень – нет проблем (красим в **черный** цвет).
- Вставка, когда отец **черный** – нет проблем.
- Если отец **красный**, то…

RB ВСТАВКА:
ОТЕЦ И ДЯДЯ КРАСНЫЕ



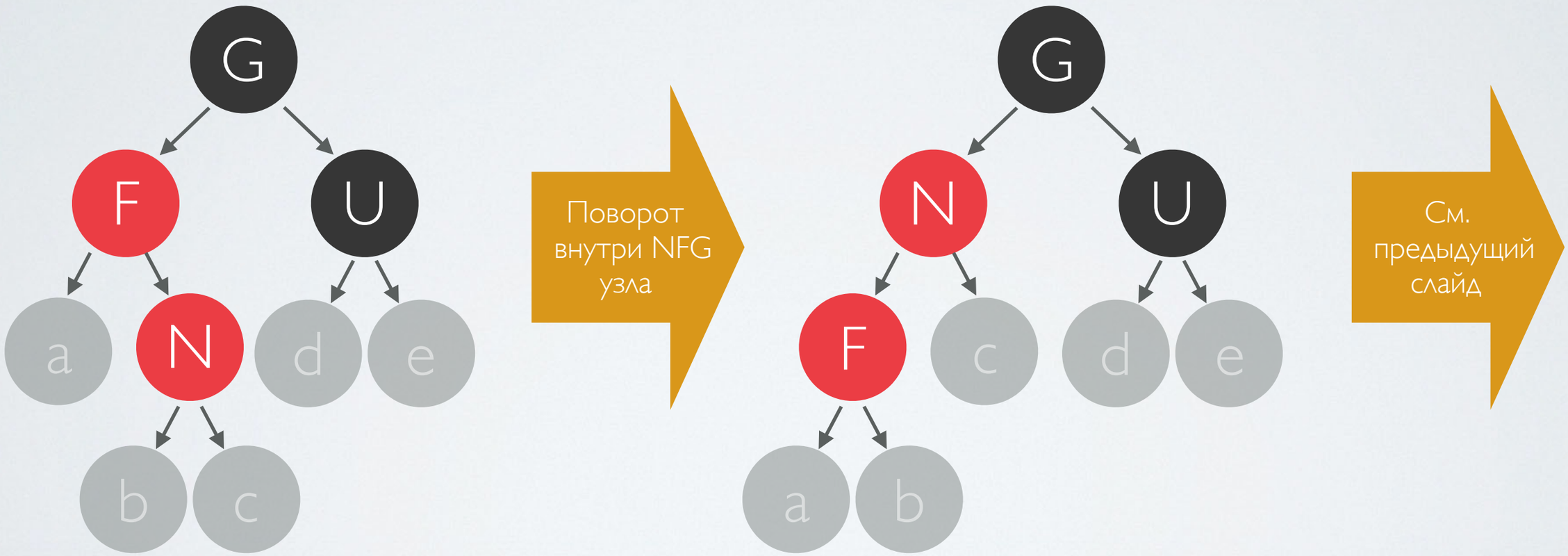
N – new, F – father, U – uncle, G - grandfather

RB ВСТАВКА:
ДЯДЯ **ЧЕРНЫЙ** (НОВЫЙ СЛЕВА)



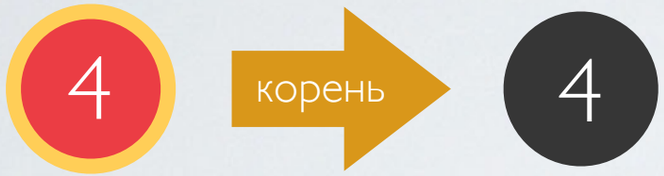
N – new, F – father, U – uncle, G - grandfather

RB ВСТАВКА:
ДЯДЯ **ЧЕРНЫЙ** (НОВЫЙ СПРАВА)

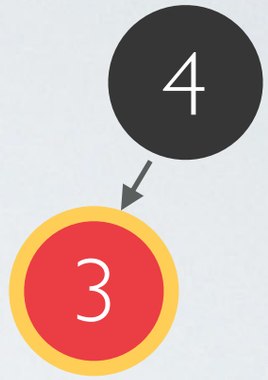


N – new, F – father, U – uncle, G - grandfather

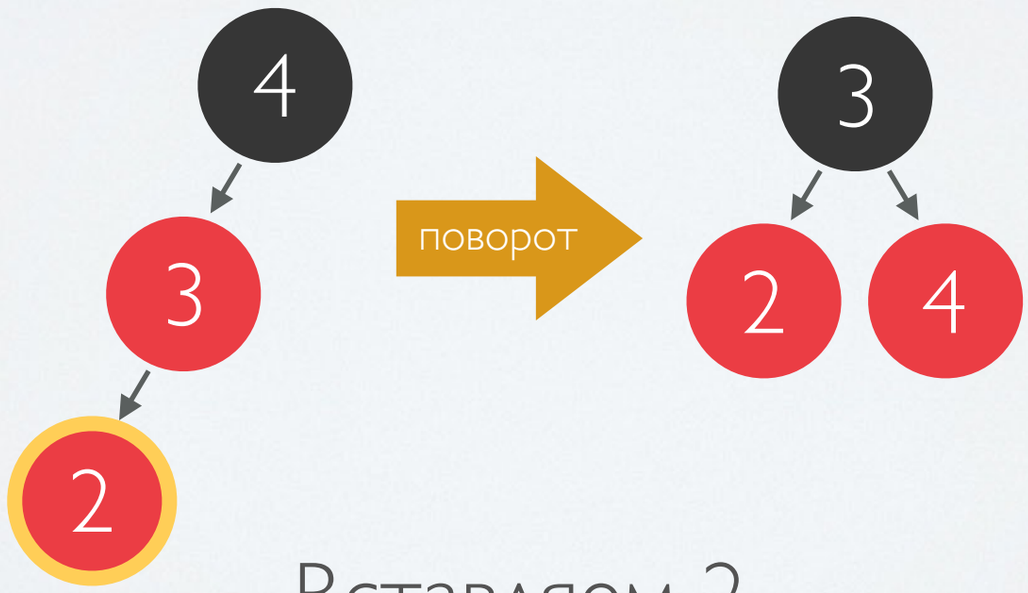
ПРИМЕР RB



Вставляем 4

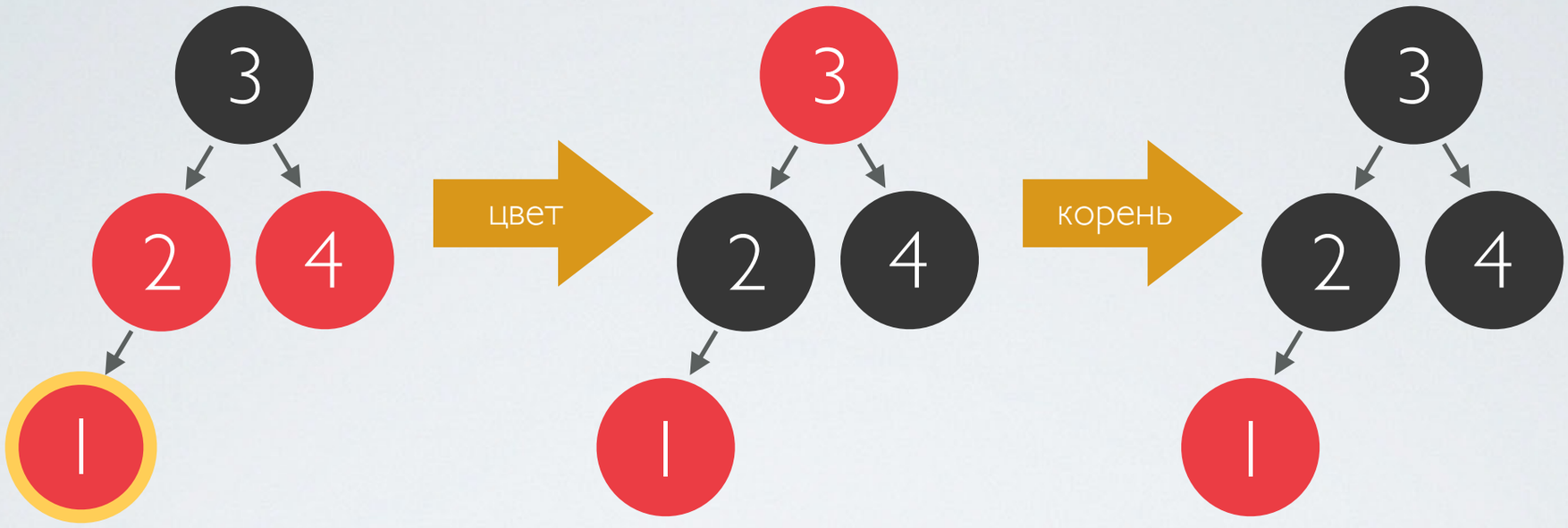


Вставляем 3

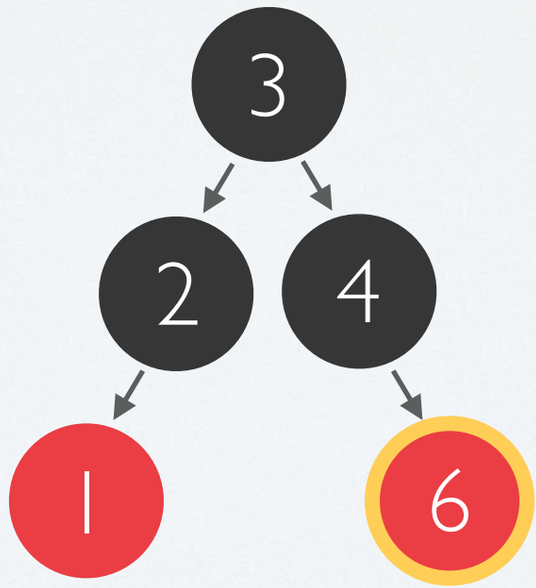


Вставляем 2

ПРИМЕР RB

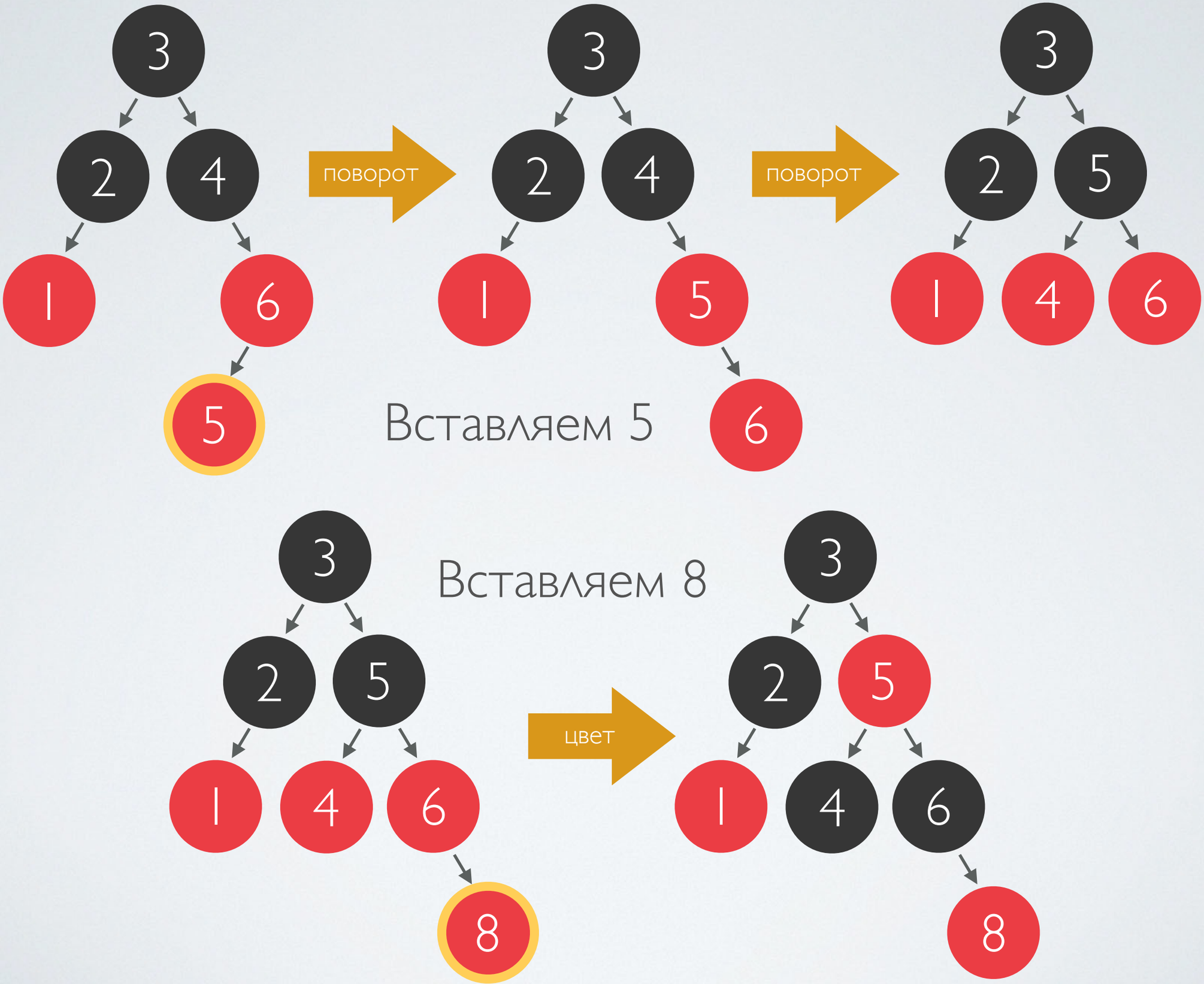


Вставляем 1



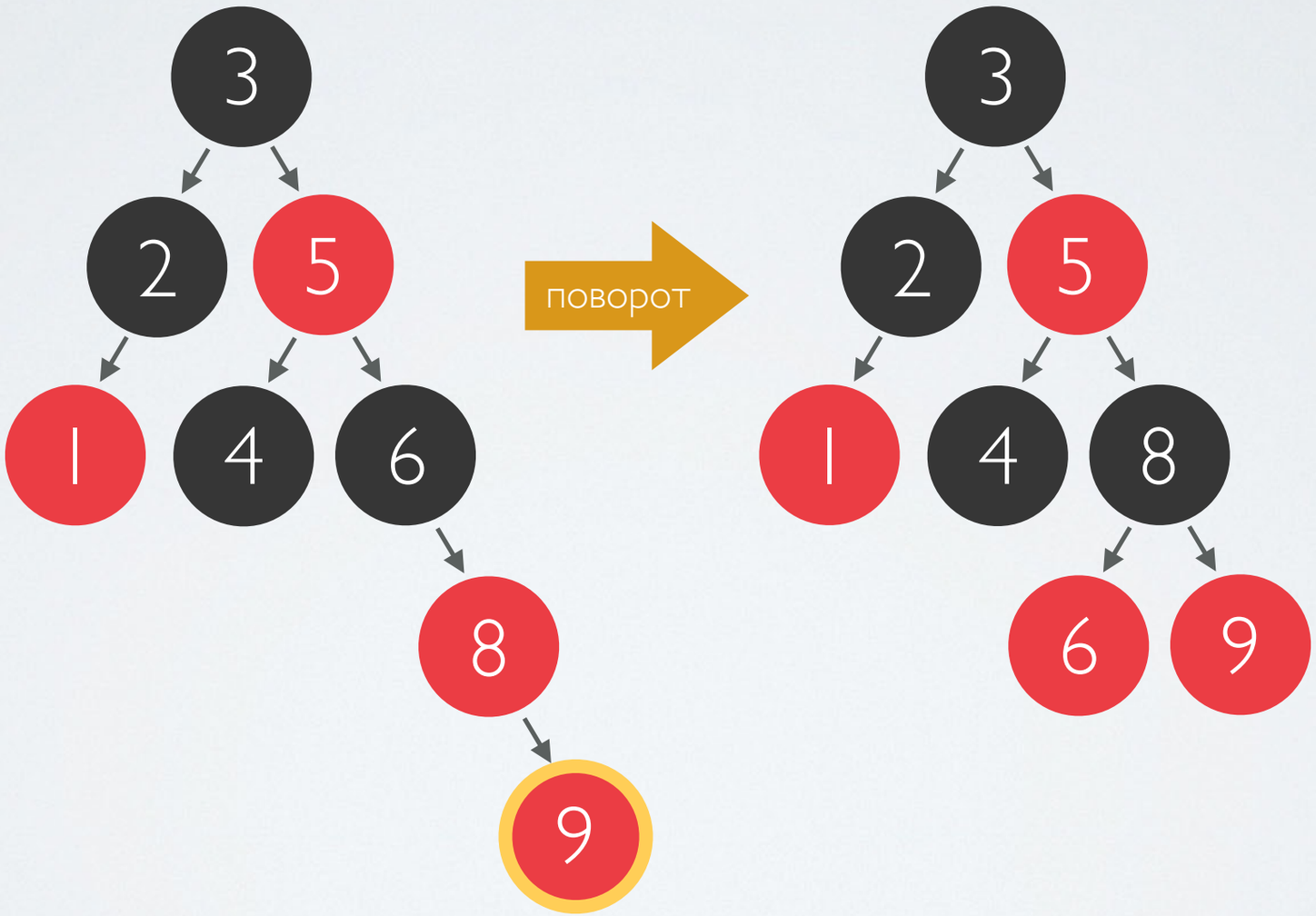
Вставляем 6

ПРИМЕР RB


3
2
4
1
6
5
поворот
3
2
4
1
5
поворот
3
2
5
1
4
6
Вставляем 5
6
Вставляем 8
3
2
5
1
4
6
8
цвет
3
2
5
1
4
6
8

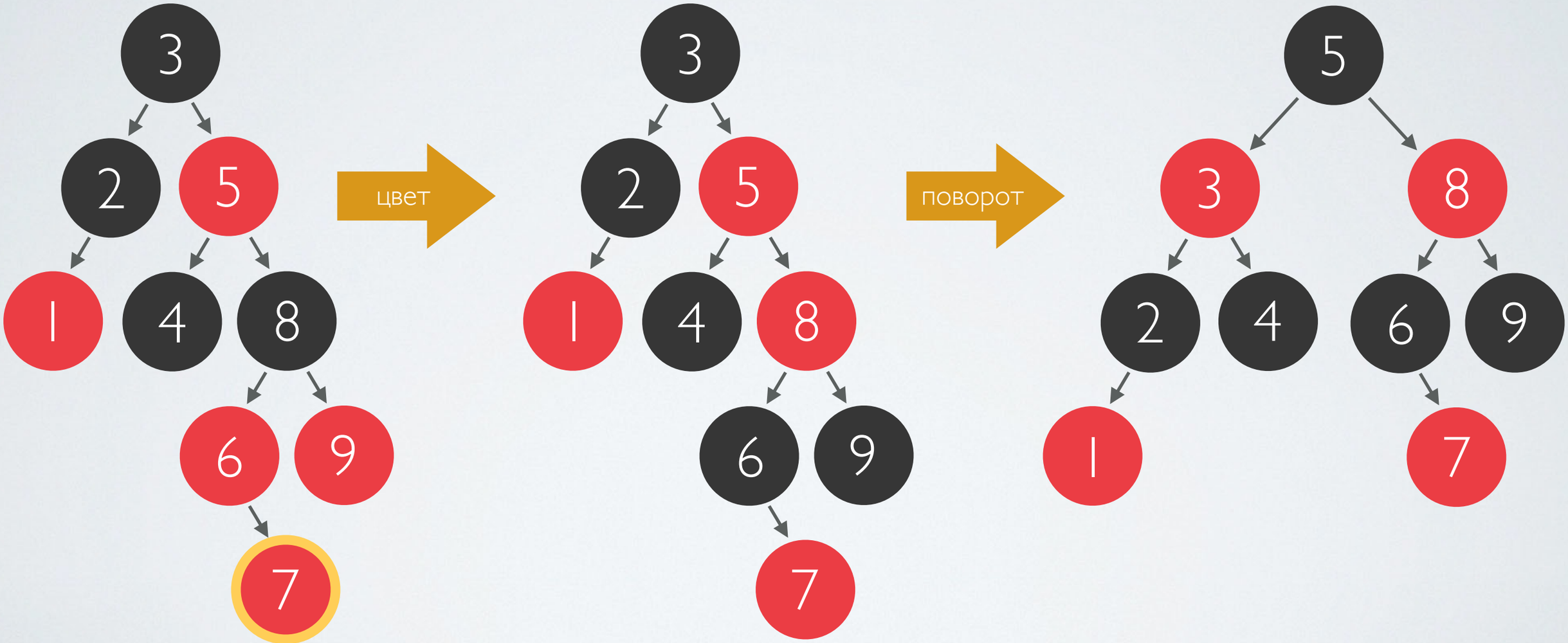
ПРИМЕР RB

Вставляем 9



ПРИМЕР RB

Вставляем 7



ЗАДАЧКИ О ДЕРЕВЬЯХ

Распишите (разрисуйте) процесс добавления в пустое...

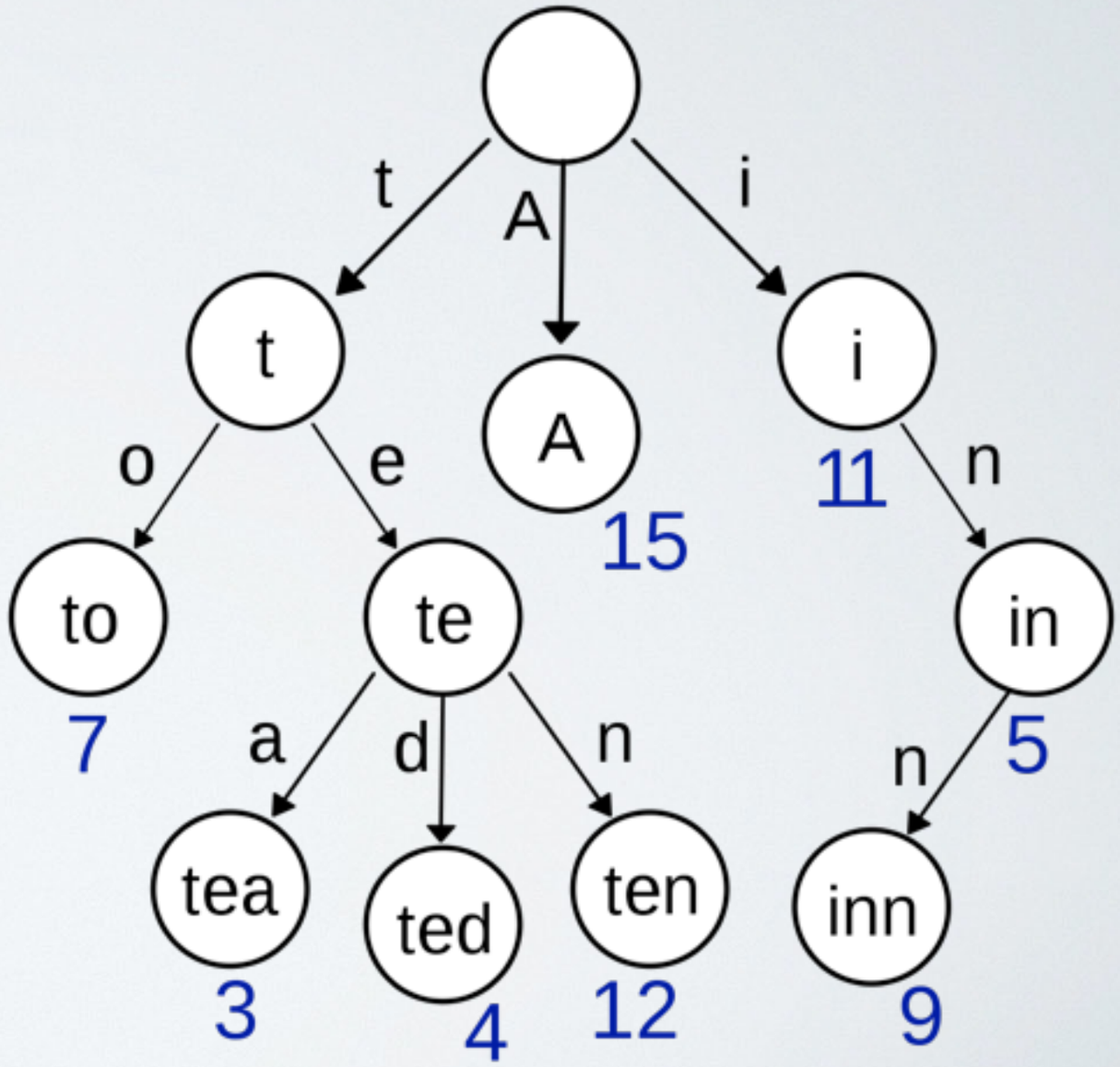
1. АВЛ-дерево значений: 1, 8, 9, 2, 5, 7.
2. 2-3-4-дерево значений: 1, 8, 4, 2, 9, 6, 7.
3. RB-дерево значений: 1, 4, 3, 8, 7, 5.

Каждая задача оценивается в 0,5 у.е.

НЕ-ДЕРЕВЬЯ И НЕ-ПОИСКА

ПРЕФИКСНОЕ ДЕРЕВО (БОР, TRIE)

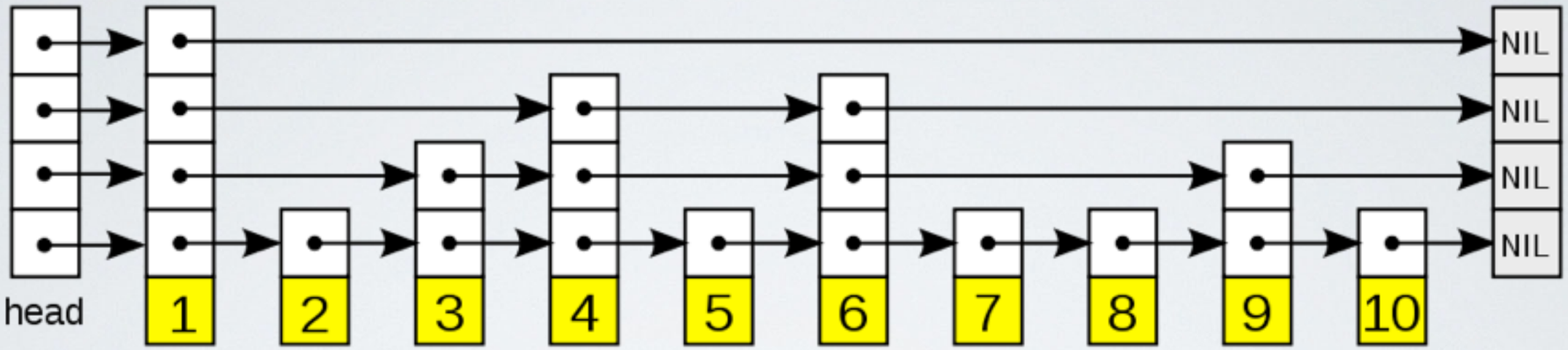
- A: 15
- to: 7
- tea: 3
- ted: 4
- ten: 12
- i: 11
- in: 5
- inn: 9



Сложность вставки и поиска?

O(длины строки)

СЛОЕНЫЕ СПИСКИ (SKIP LIST)



- Нижний список всегда содержит все элементы.
- Вероятность попадания в список уровнем выше – **p** (**p=1/2** или **1/4**). Еще выше – p^2 и т.д. (кидаем монету).
- При удалении элемента удаляем его из всех списков.

КОНЕЦ СЕДЬМОЙ ЛЕКЦИИ