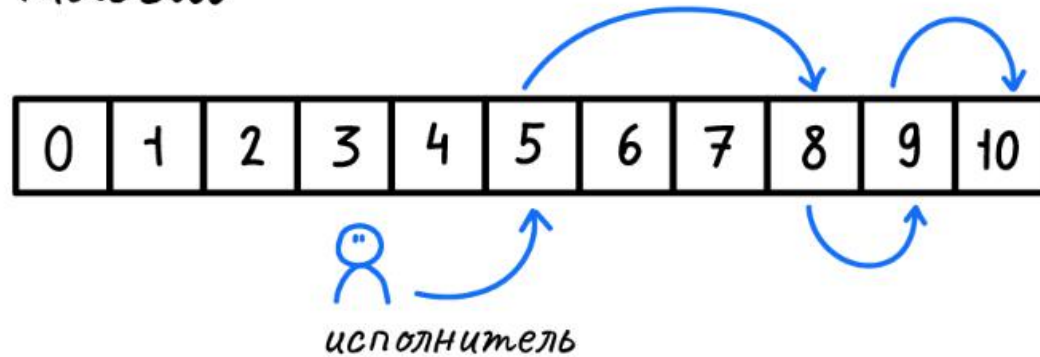


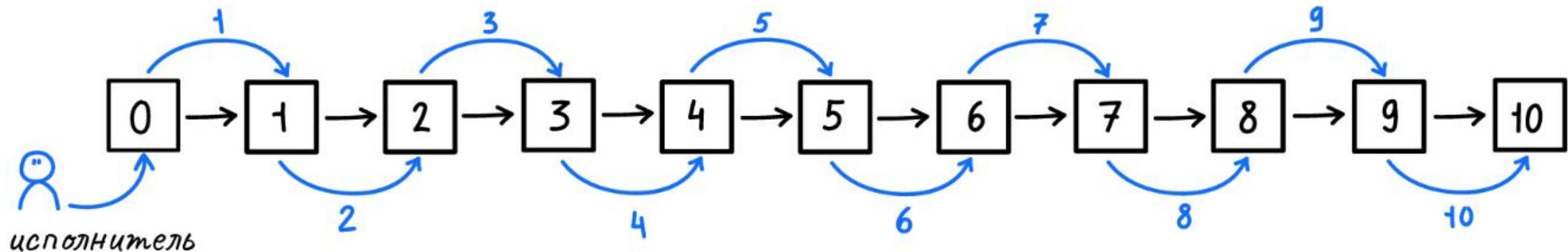
Красно-чёрное дерево

Рассмотрим ситуацию поиска числа в отсортированном массиве уникальных чисел

массив



список

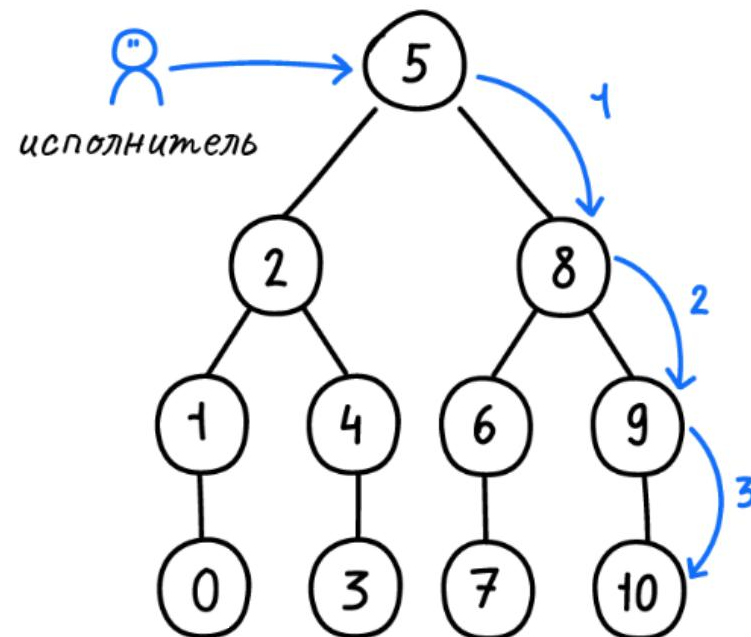


Бинарное дерево поиска

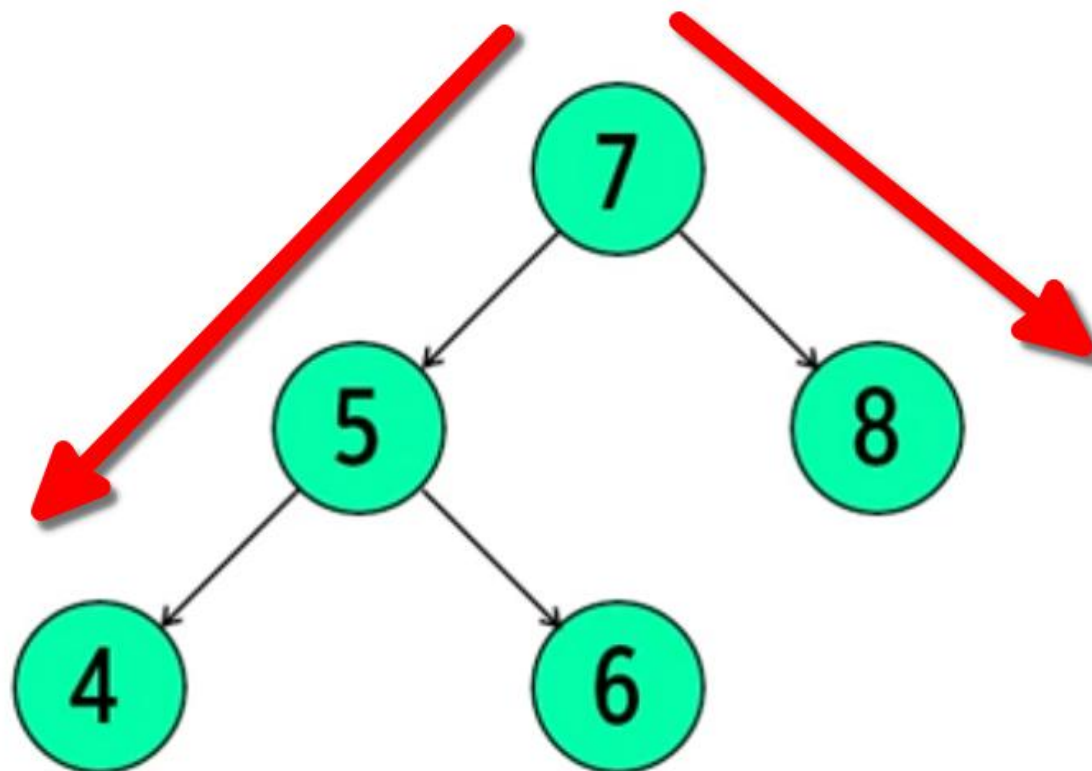
Бинарное дерево поиска - бинарное, дерево, для которого выполнены условия:

- оба поддерева — левое и правое — являются двоичными деревьями поиска;
- у всех узлов *левого* поддерева произвольного узла X значения ключей данных **меньше либо равны**, нежели значение ключа данных самого узла X;
- у всех узлов *правого* поддерева произвольного узла X значения ключей данных **больше**, нежели значение ключа данных самого узла X.

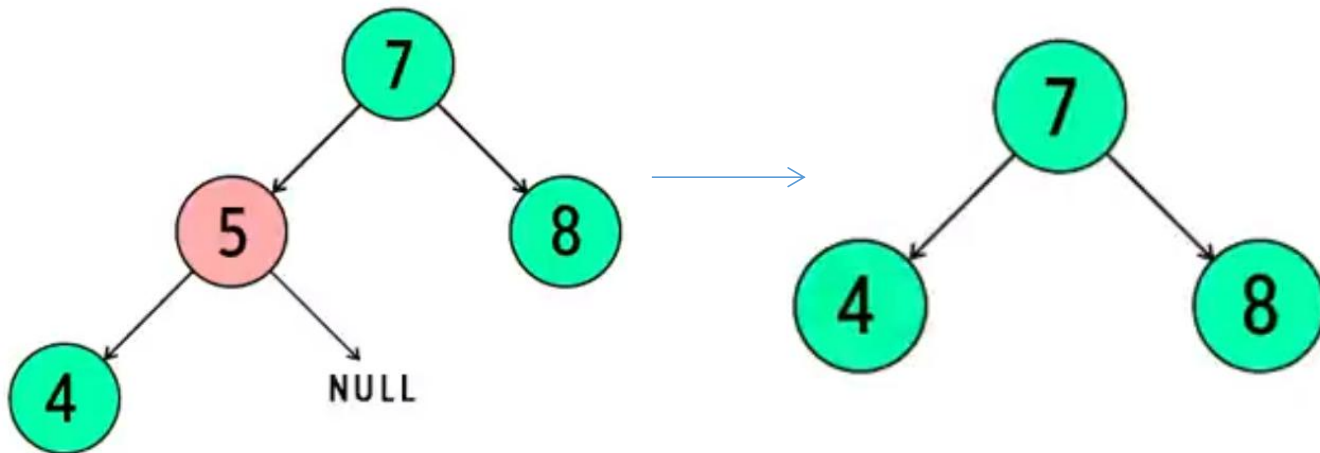
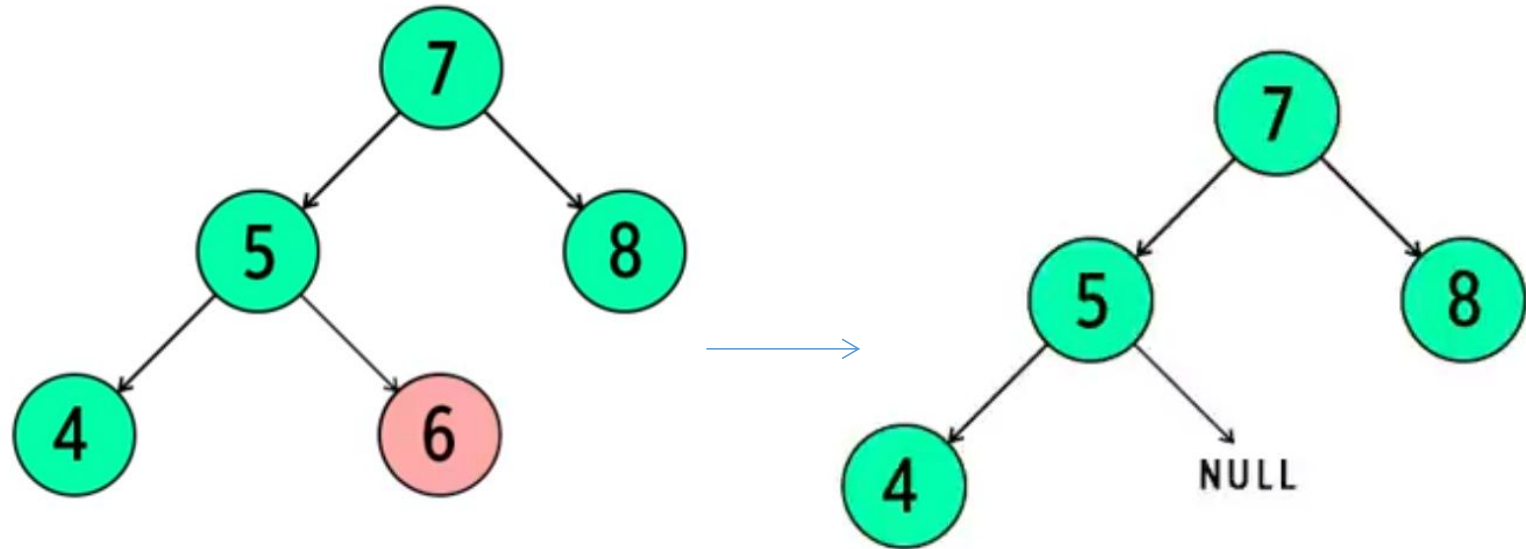
бинарное дерево поиска



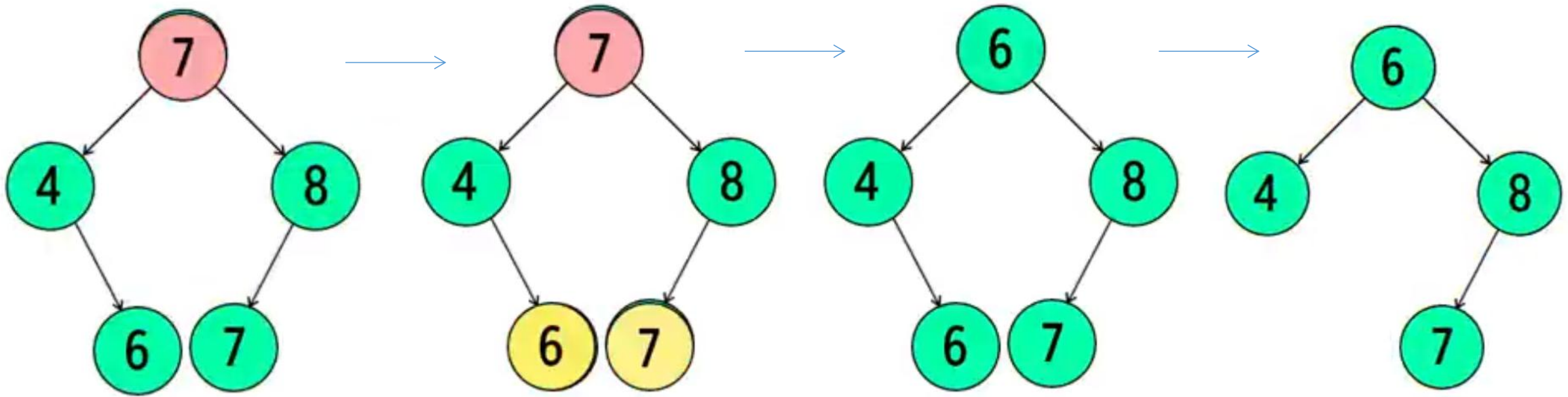
Поиск минимума и максимума



Удаление узлов в случае листов или одного ребёнка



Удаление узлов в случае двух детей

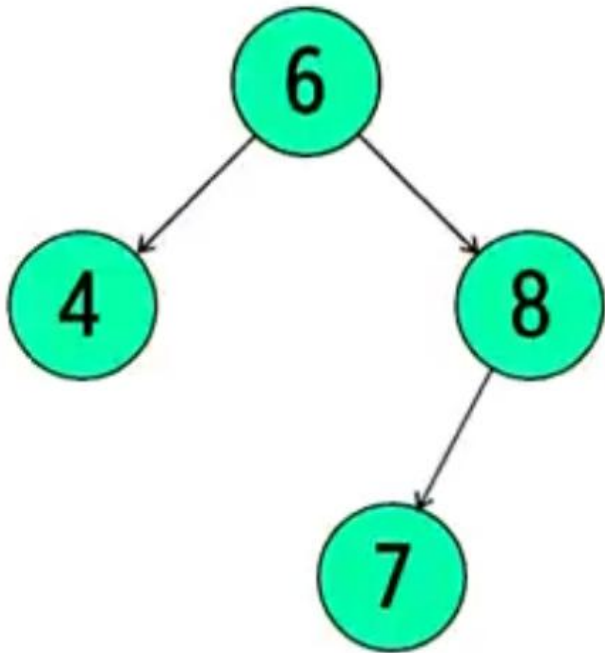


должны найти либо наибольший из левой ветки либо наименьший из правой

Обходы деревьев: симметричный

Чтобы проверить, что все элементы остались на месте, кроме того, который мы хотели удалить, совершим обход:

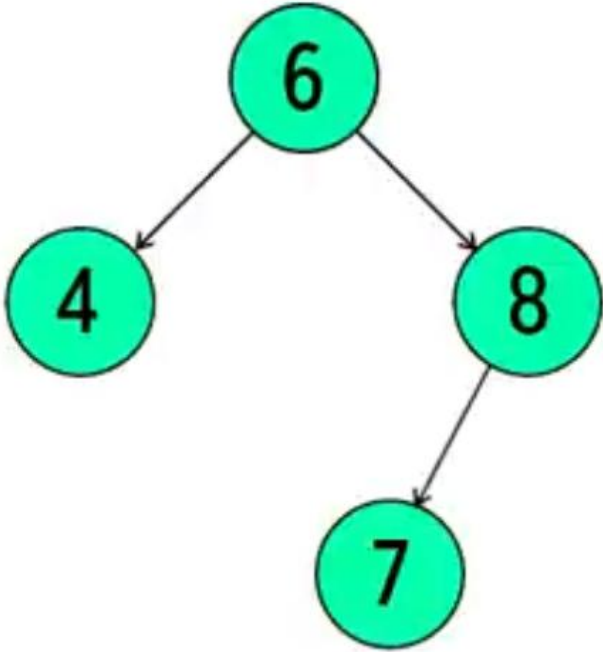
левый - родитель - правый



Обходы деревьев: **обратный**

Для чего может быть нужен следующий обход :

левый - правый - родитель
правый - левый - родитель ?



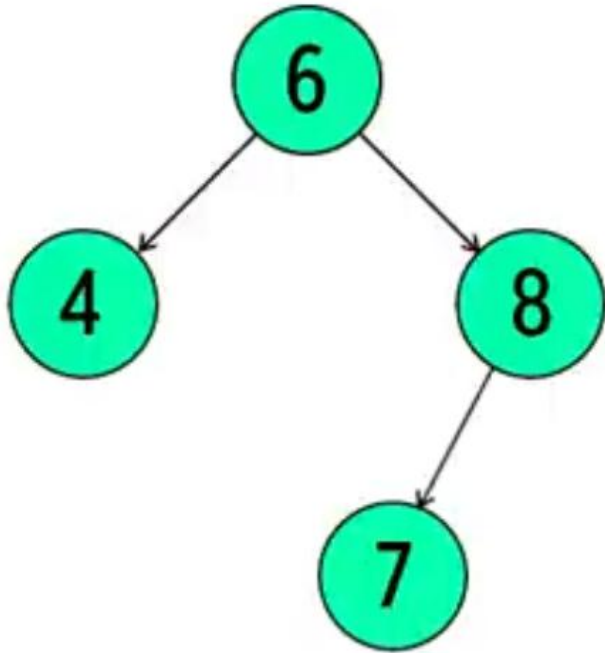
Для очистки памяти.



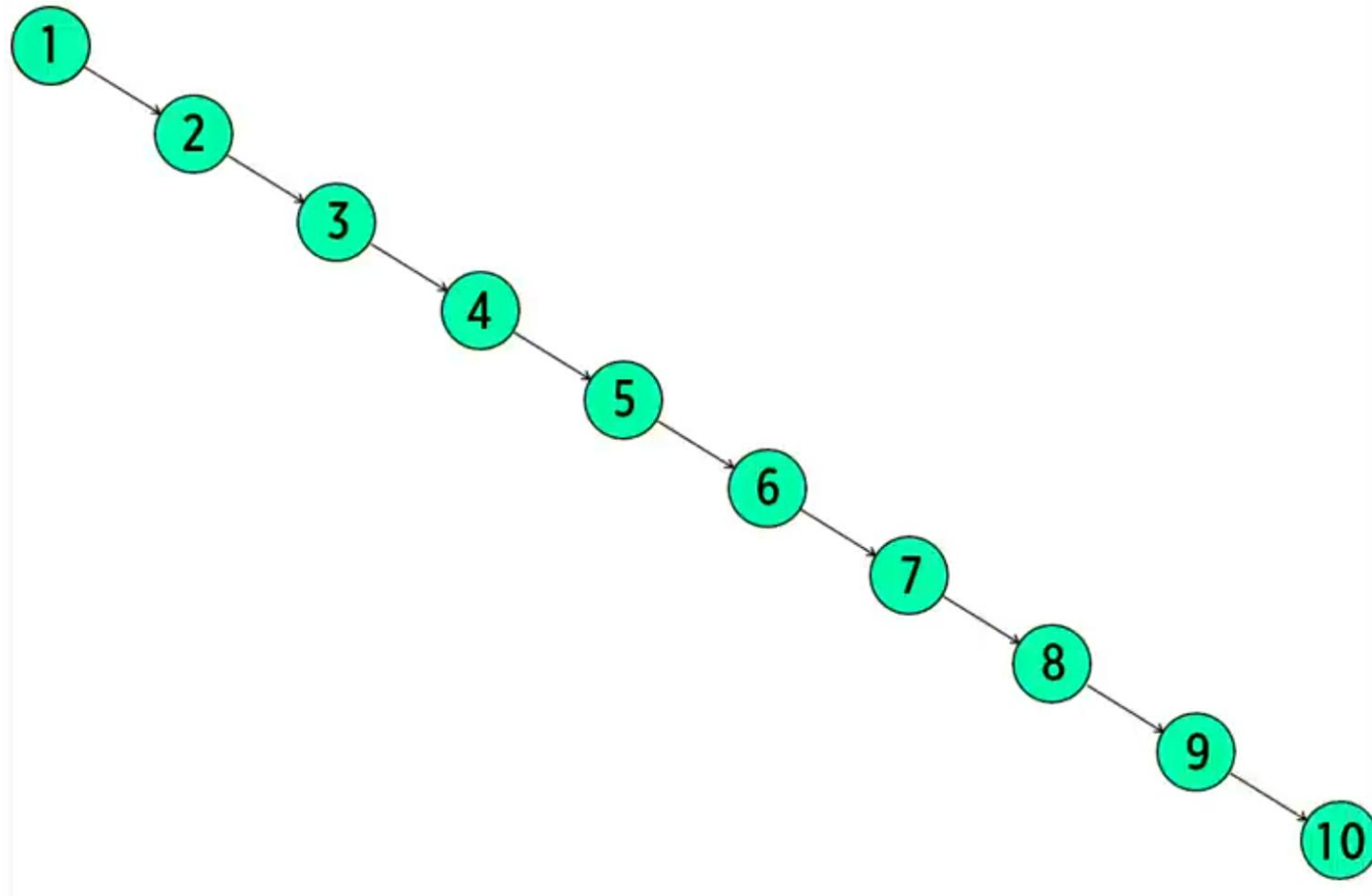
Обходы деревьев: **прямой**

родитель - левый - правый

Может быть полезен при копировании дерева.

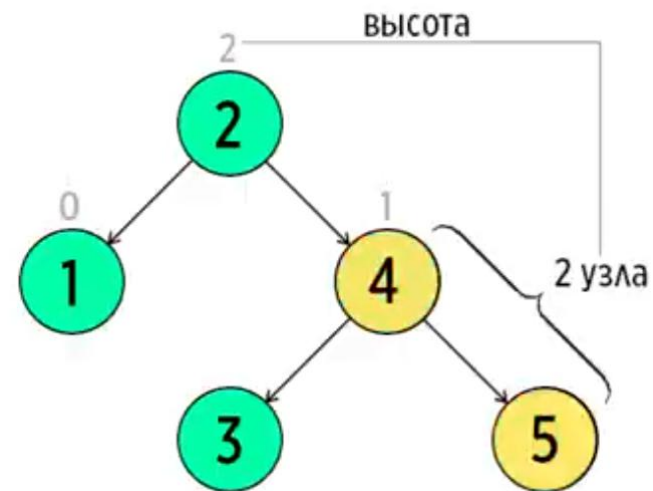
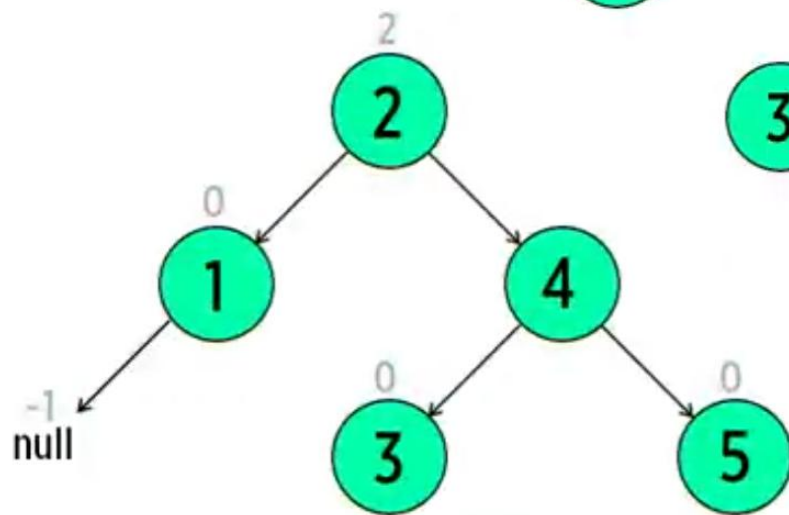
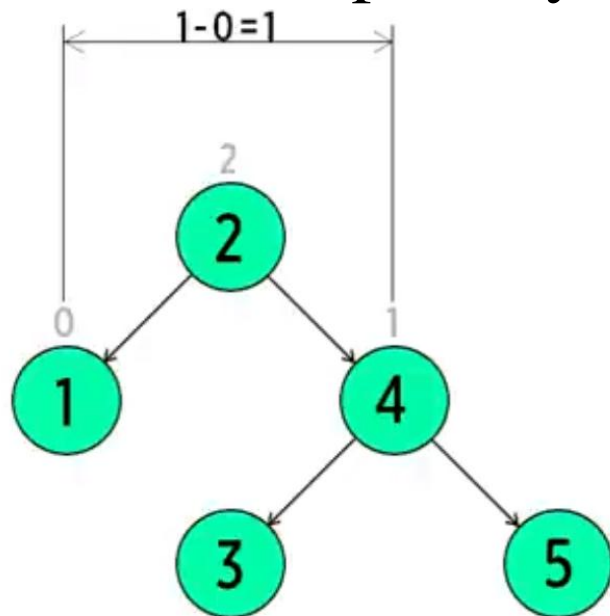


Балансировка



Балансировка

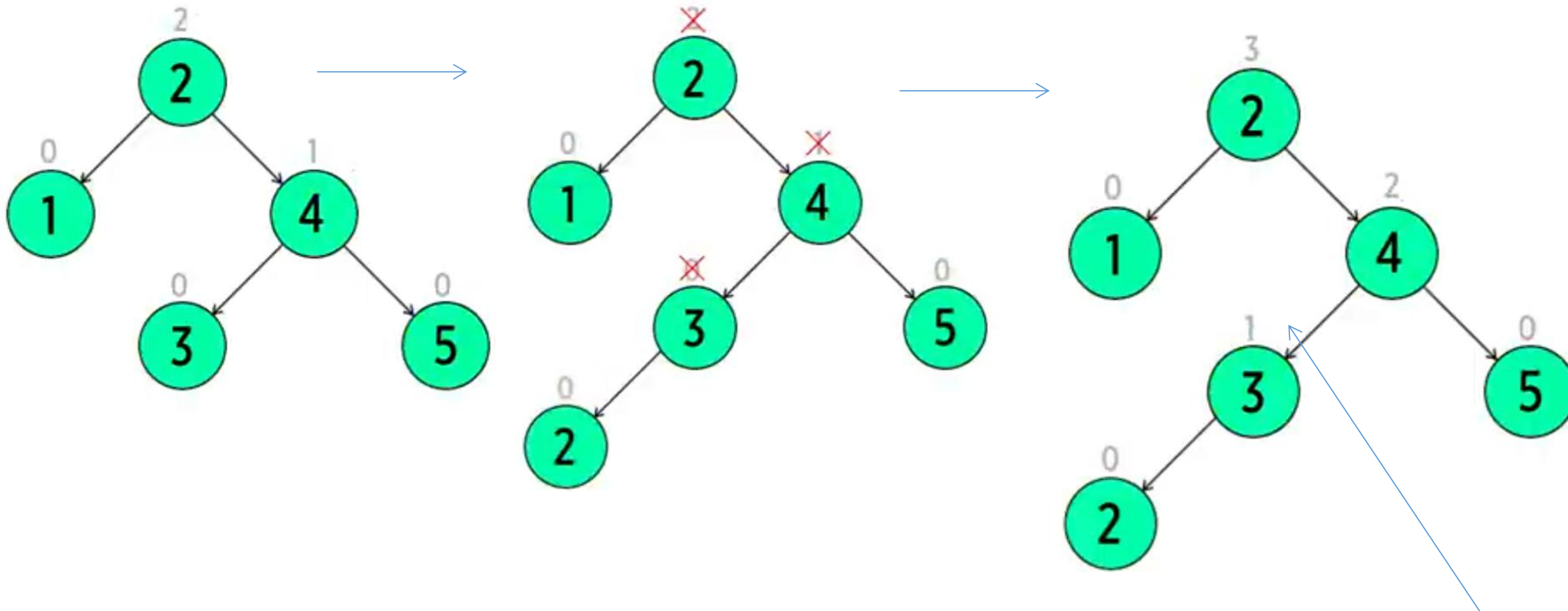
Задача сделать так, чтобы разница между
высотами каждого дочернего узла была
 $|r| < 1$



Высота узла - длина пути от ребёнка
до самого нижнего узла

Изменение высот при добавлении вершин

Новая высота = $\max(\text{height}(\text{left}), \text{height}(\text{right})) + 1$



Если есть 1 ребёнок, то при добавлении второго высота не меняется

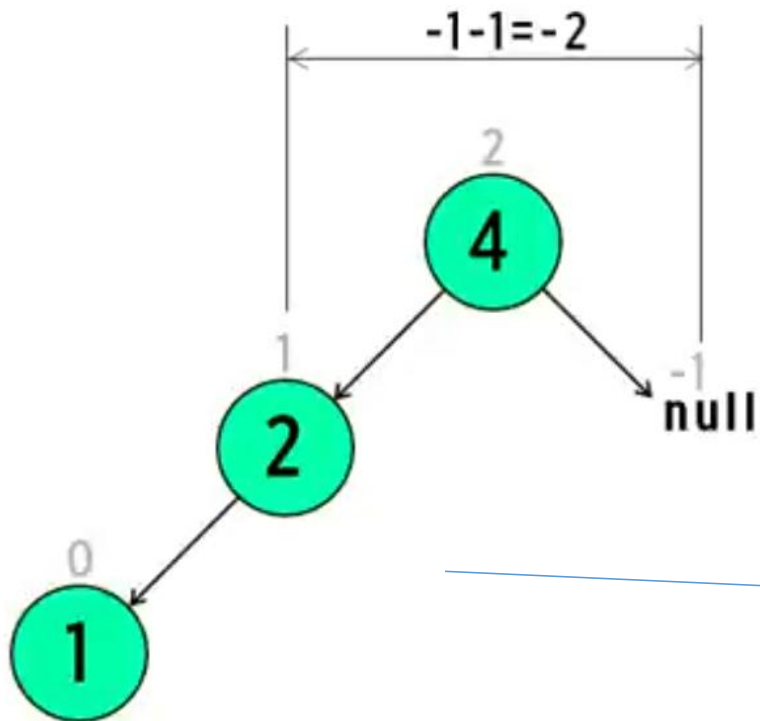
$$\max(0, -1) + 1 = 0 + 1 = 1$$

Перегруженность влево

Вычитаем из высоты правого узла высоту левого.

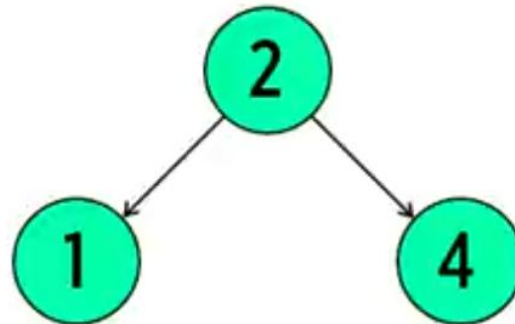
Если < 0 , то перегруженность слева.

Если > 0 , то справа.



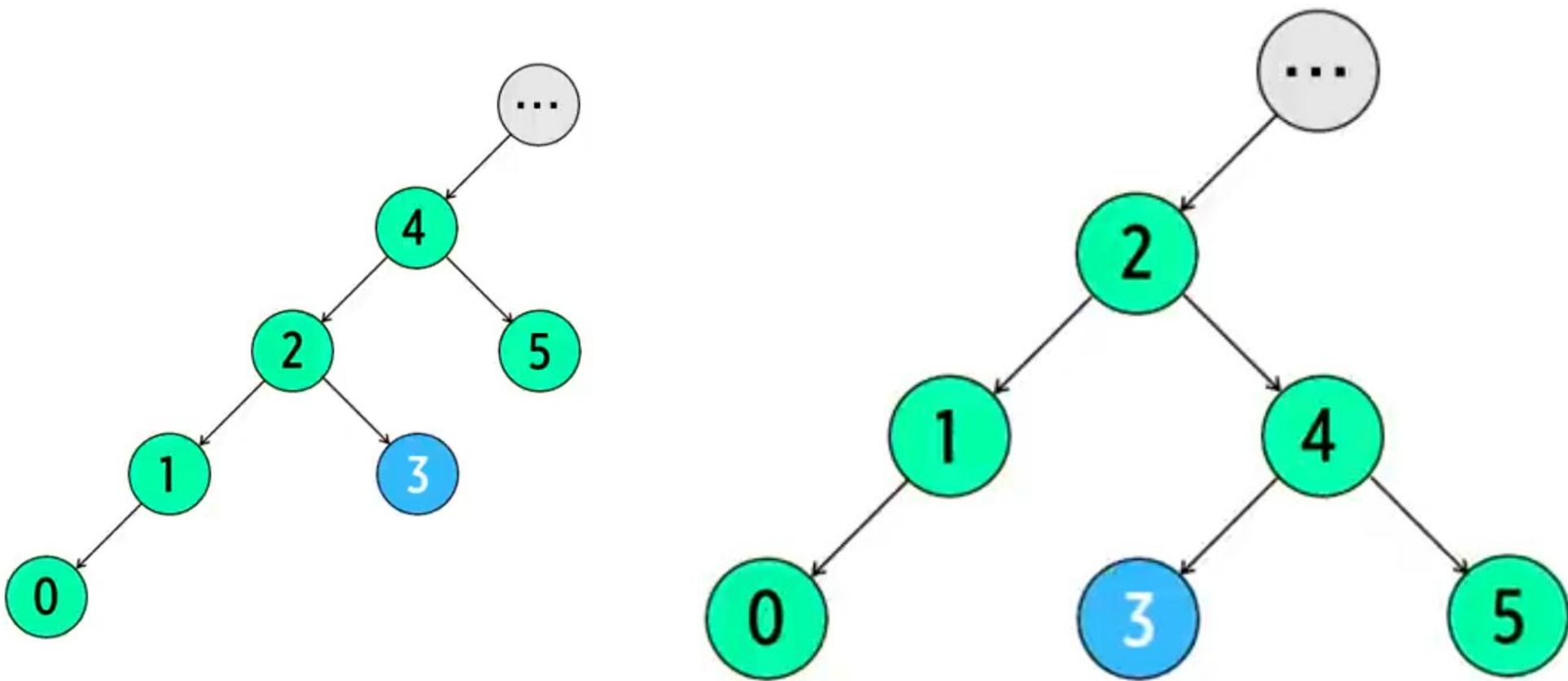
Правый поворот

Если изначальный узел, относительно которого мы делали правый поворот, указывал на левого ребёнка. То после поворота ребёнок указывает на родителя СПРАВА.



Правый поворот

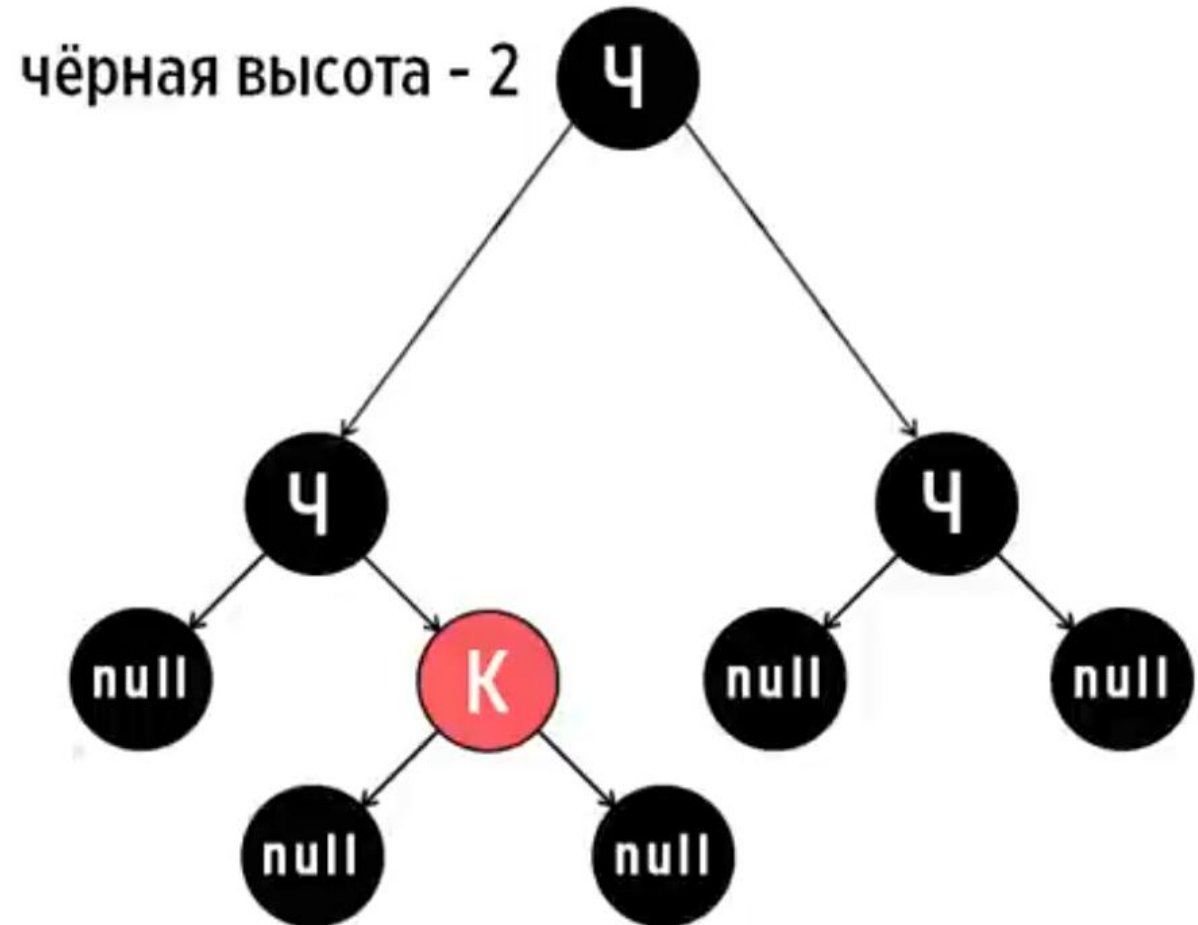
делаем swap, а затем отсоединяем 5



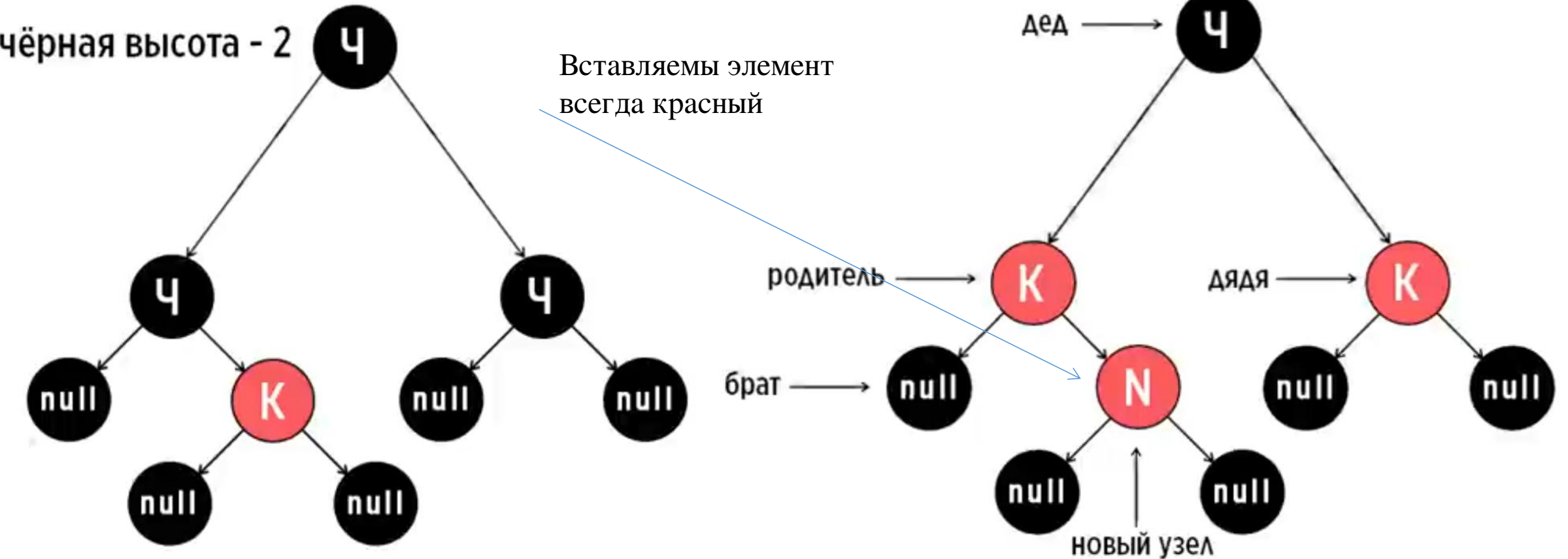
Красно-чёрное дерево

- Листом является не свободный узел со значением, а NULL
- Корень и листья - чёрные.
- У красных узлов оба ребёнка чёрные.

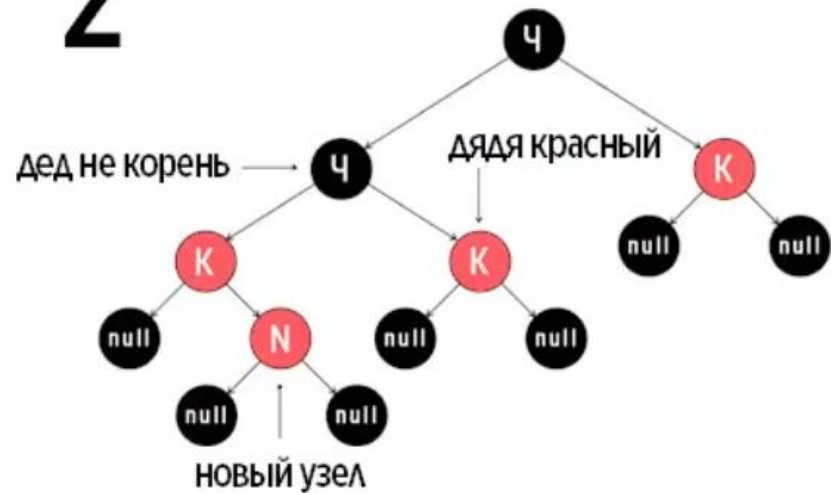
Чёрная высота - количество узлов на любом пути от корня, не включая его самого.



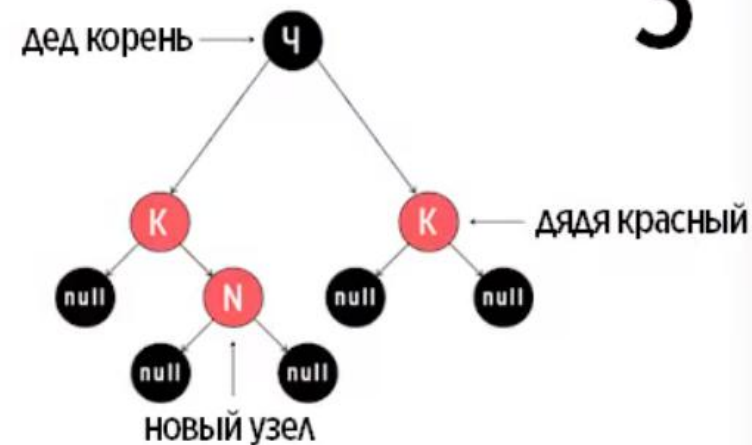
Вставка нового элемента.



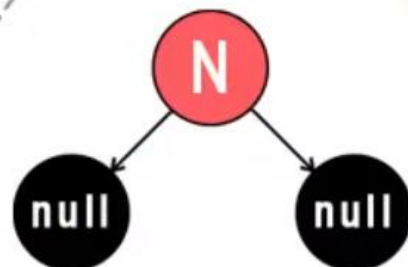
2



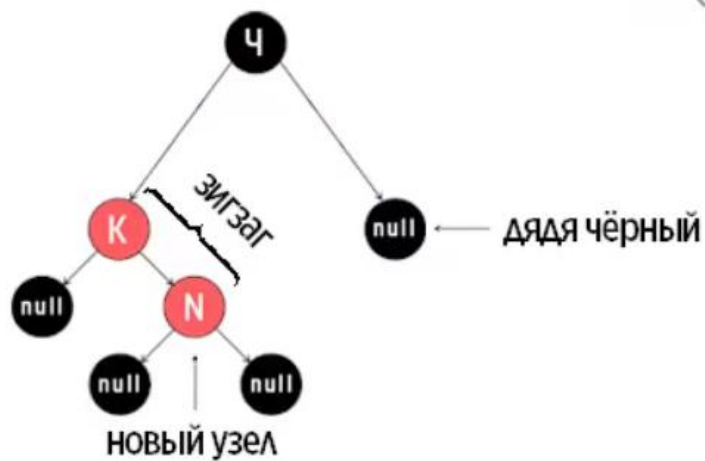
3



новый корень



4



5

