



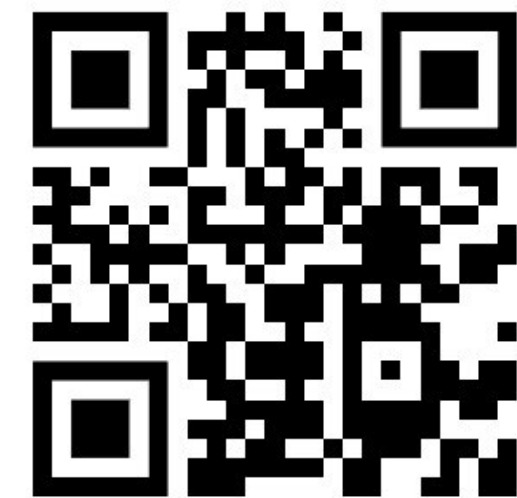
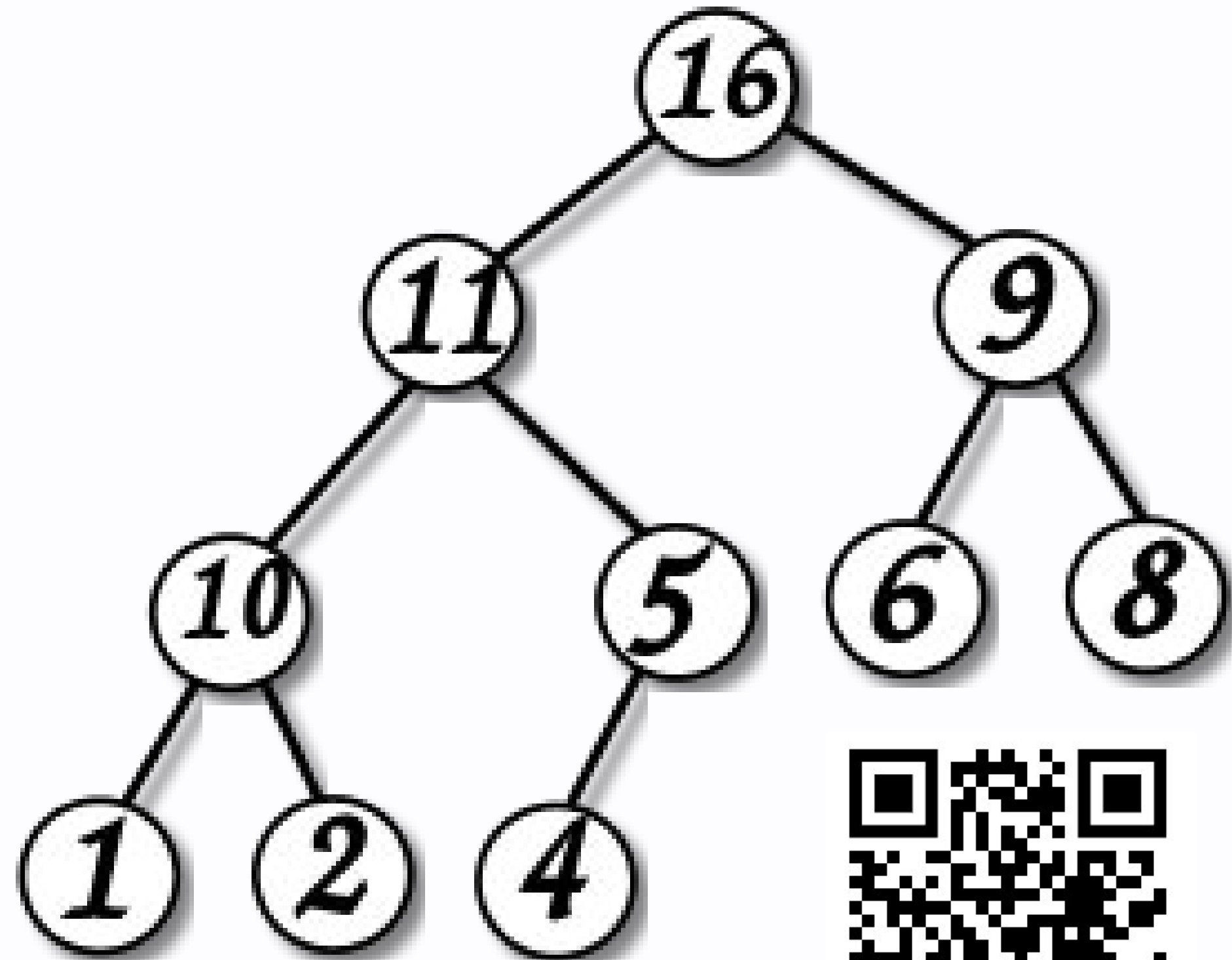
Деревья



Куча

Куча –

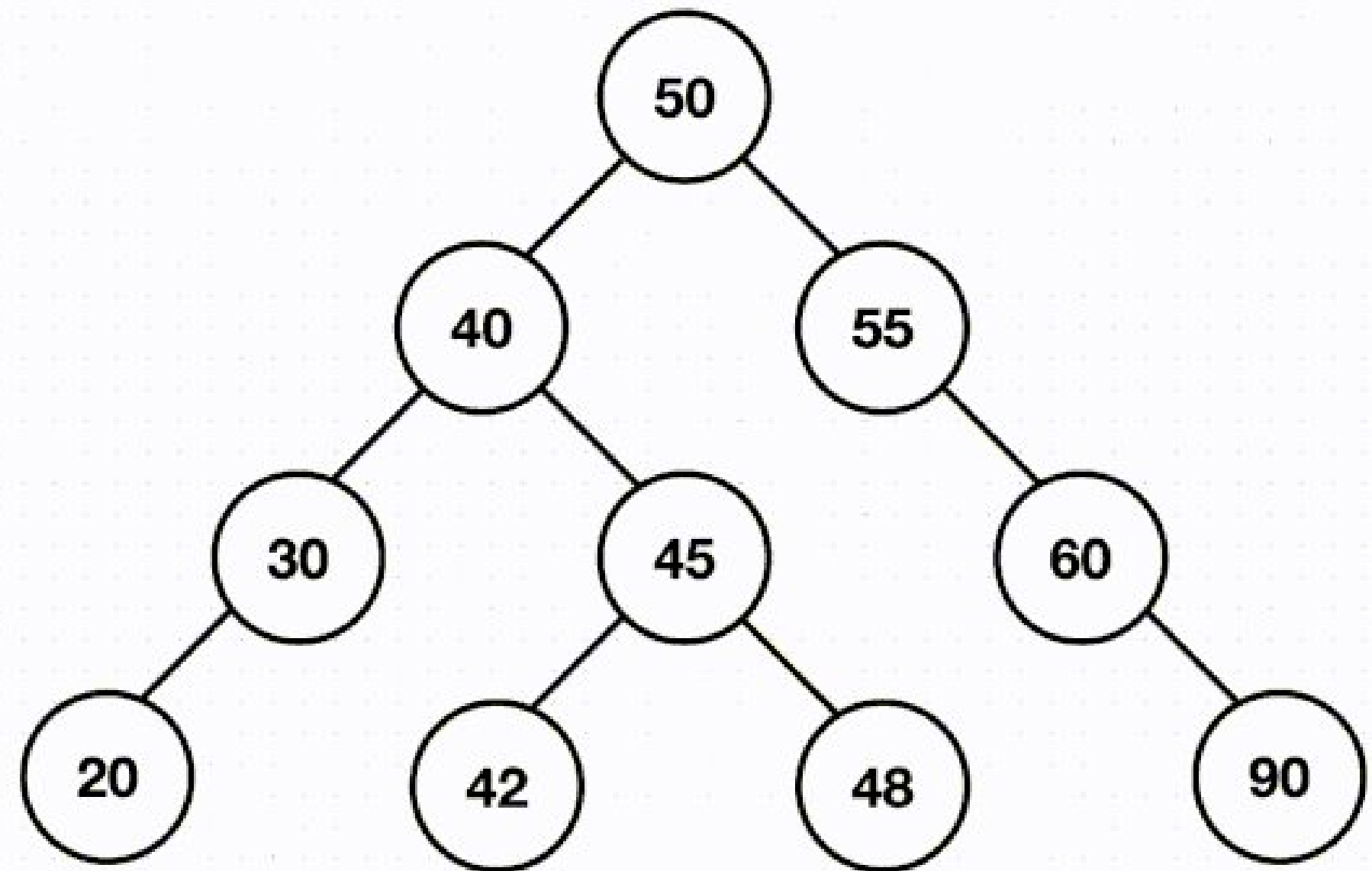
очень популярная структура на собеседованиях. Позволяет нам добавлять, удалять и искать минимуму за $O(\log n)$



BST

Binary Search Tree –

очень популярная структура на собеседованиях. Позволяет нам добавлять, удалять и искать элементы за $O(\log n)$



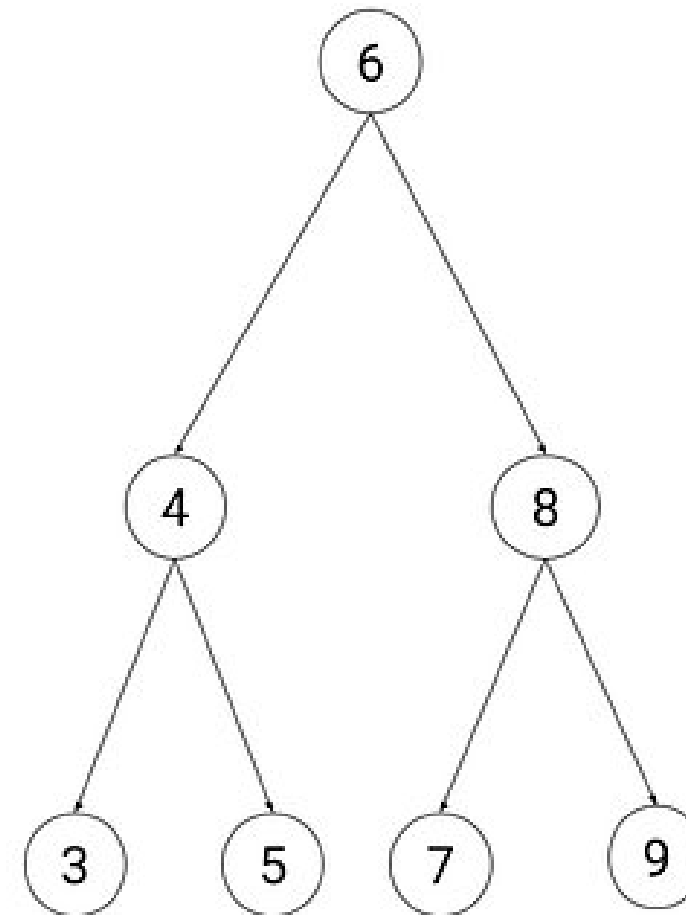
Основные свойства BST



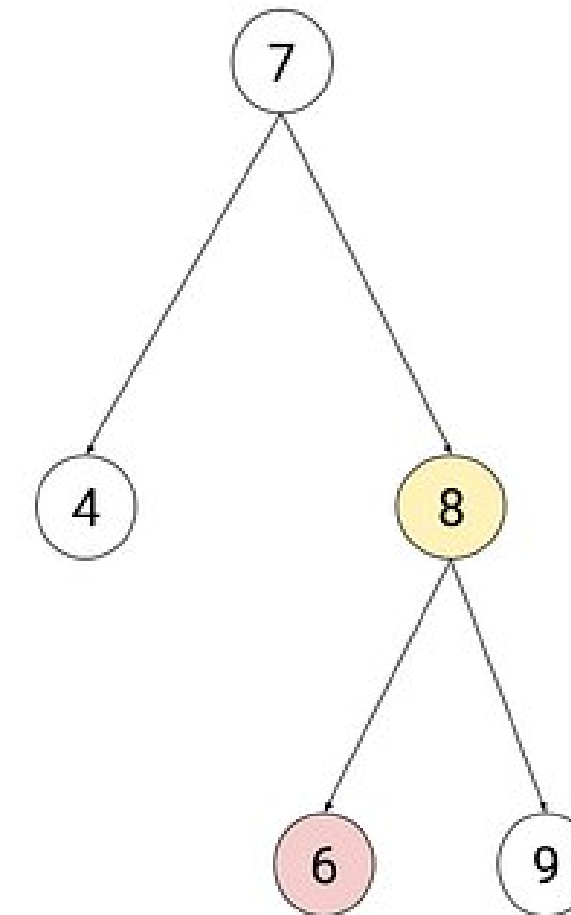
У каждой вершины может быть один, два или ни одного сына.



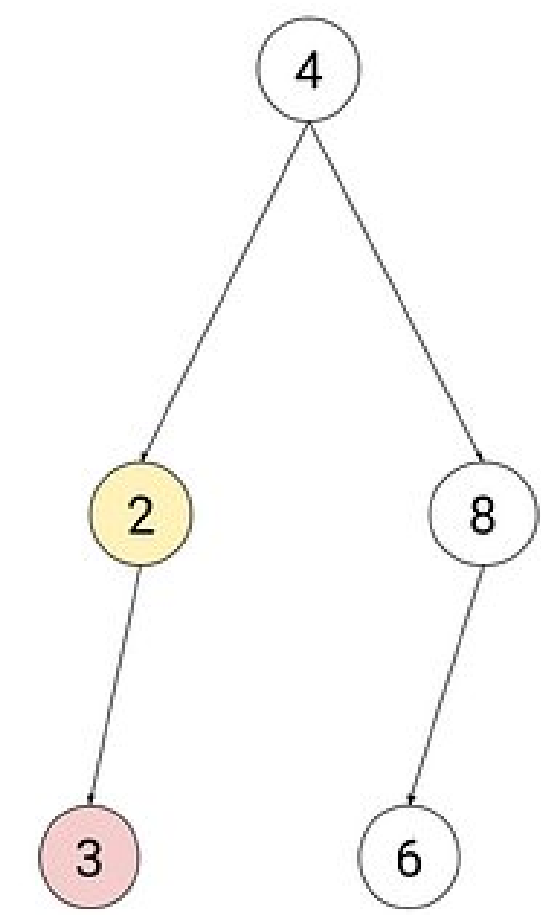
Все вершины в левом поддереве меньше, чем значение в вершине, а в правой больше, это условие выполняется для каждой вершины.



A valid BST



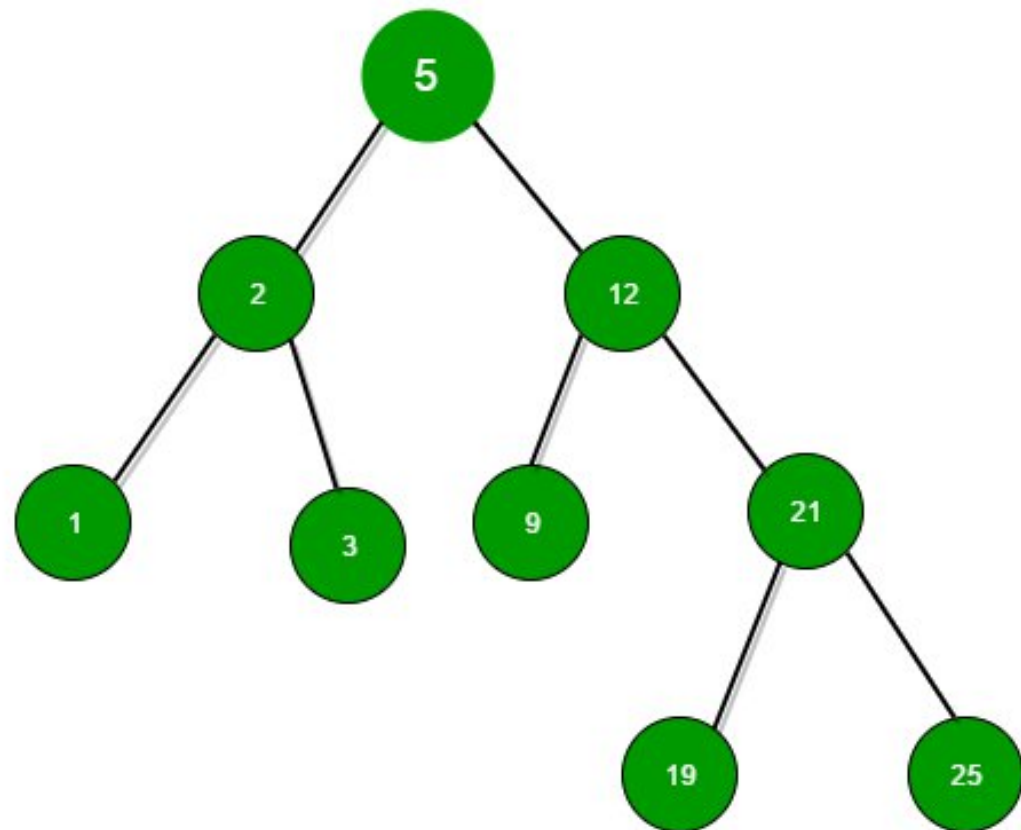
Not a valid BST



Not a valid BST

Поиск вершины в BST

```
def find(node, x):  
    if node.val == x:  
        return node  
    if node.val < x:  
        return find(node.r, x)  
    return find(node.l, x)
```



Какая сложность у этого поиска?
В лучшем, в худшем случае?



Как на поиск влияет
структура дерева?



Как построить BST
из отсортированного массива?

**Как можно
баланси́ровать BST?**



Как можно баланси́ровать BST?



AVL



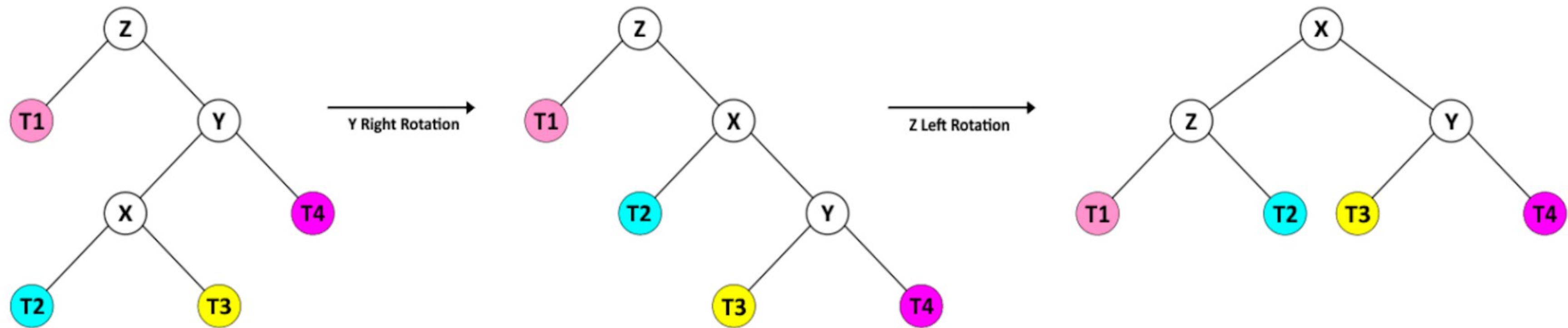
Red-black tree



Treap
(Куча+Дерево)

AVL

Основное свойство — у любой вершины глубина левого и правого поддеревьев отличается не более, чем на 1. Чтобы поддерживать это свойство, нужно его переподвешивать



AVL



За сколько работает
поиск в этом дереве?



Как доказать эту сложность?



Сколько дополнительной памяти
нужно для хранения информации о
вершине?

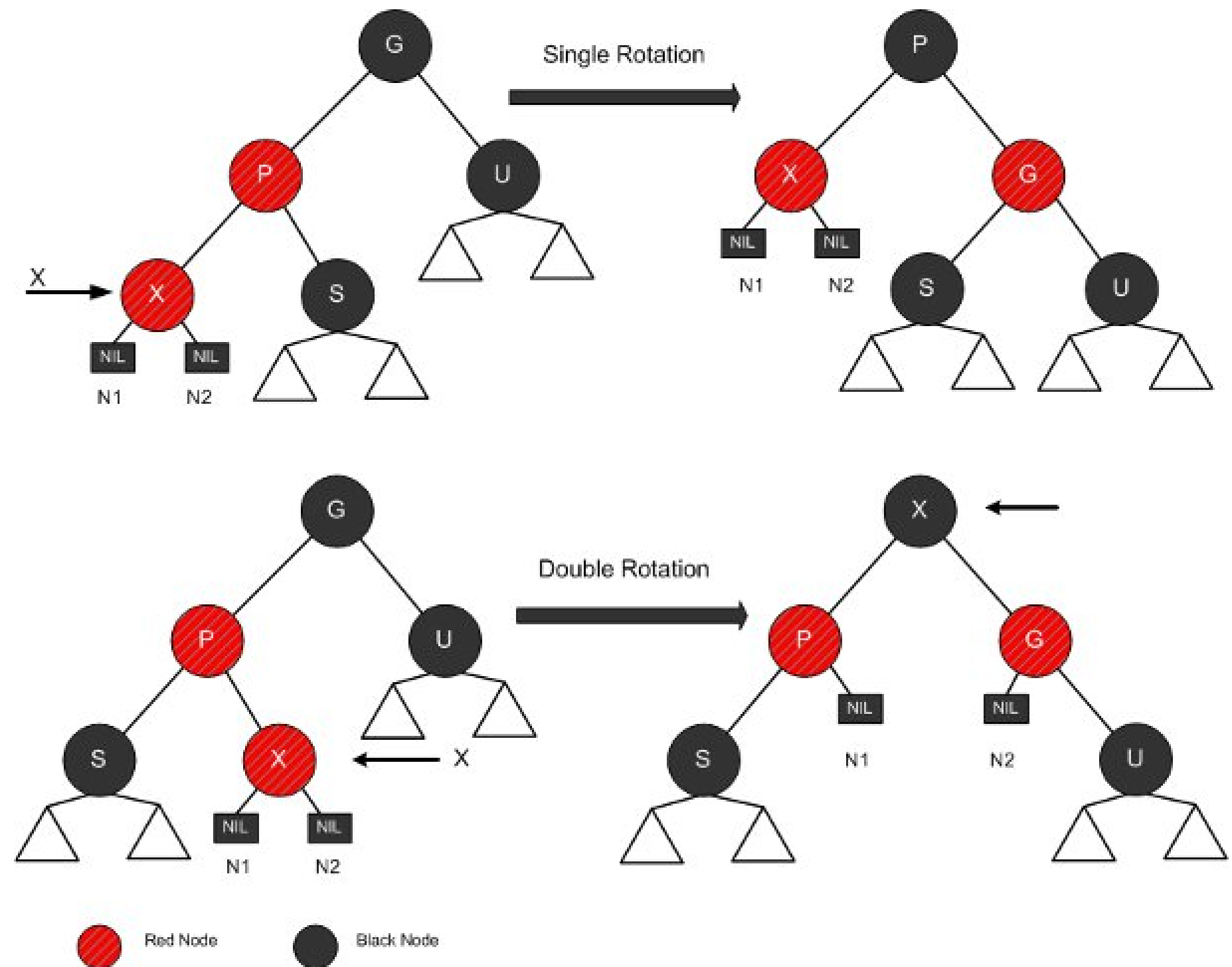
Red-black tree

Вершины имеют цвет (как думаете какие?), и на цвета наложены некоторые правила:

- На пути от корня до любого листа одинаковое количество черных вершин
- У красной вершины нет красных детей



Может ли дерево состоять только из вершин одного цвета? При каких условиях?

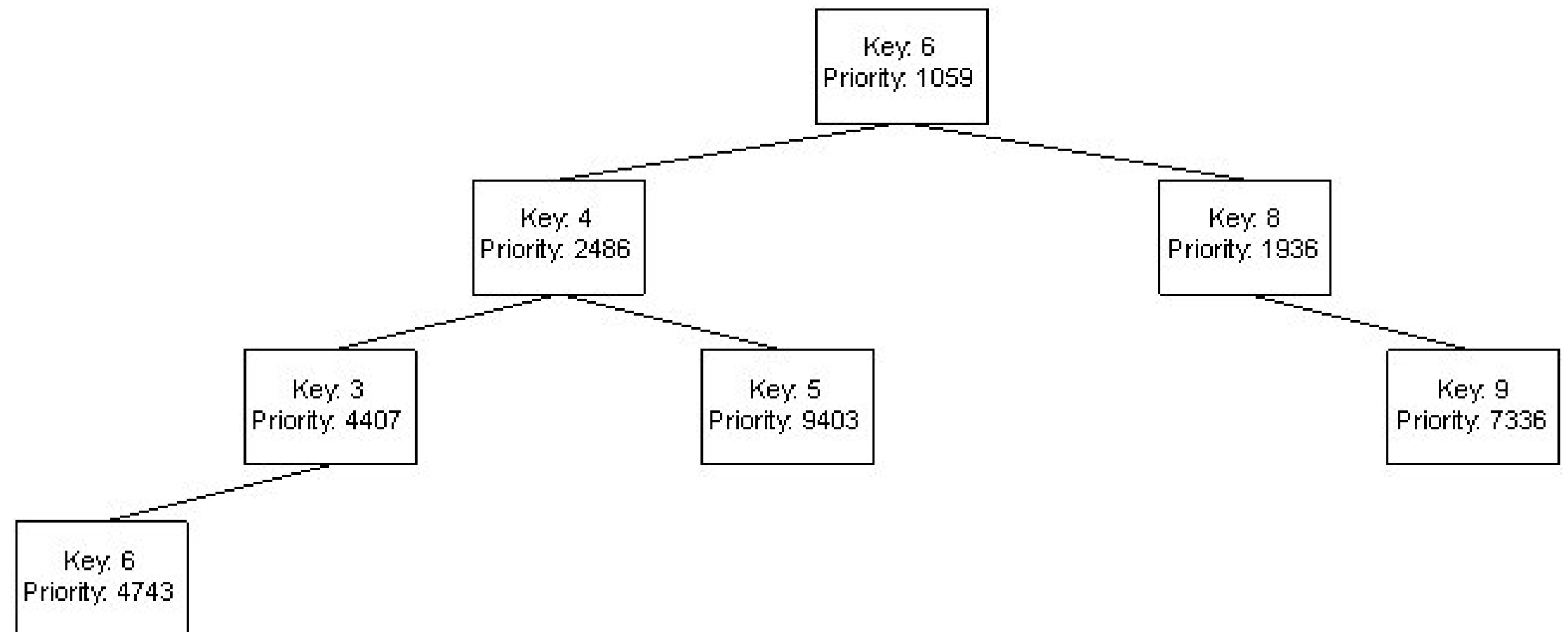


Декартово дерево

i

Смесь кучи и BST. У каждой вершины есть два значения.

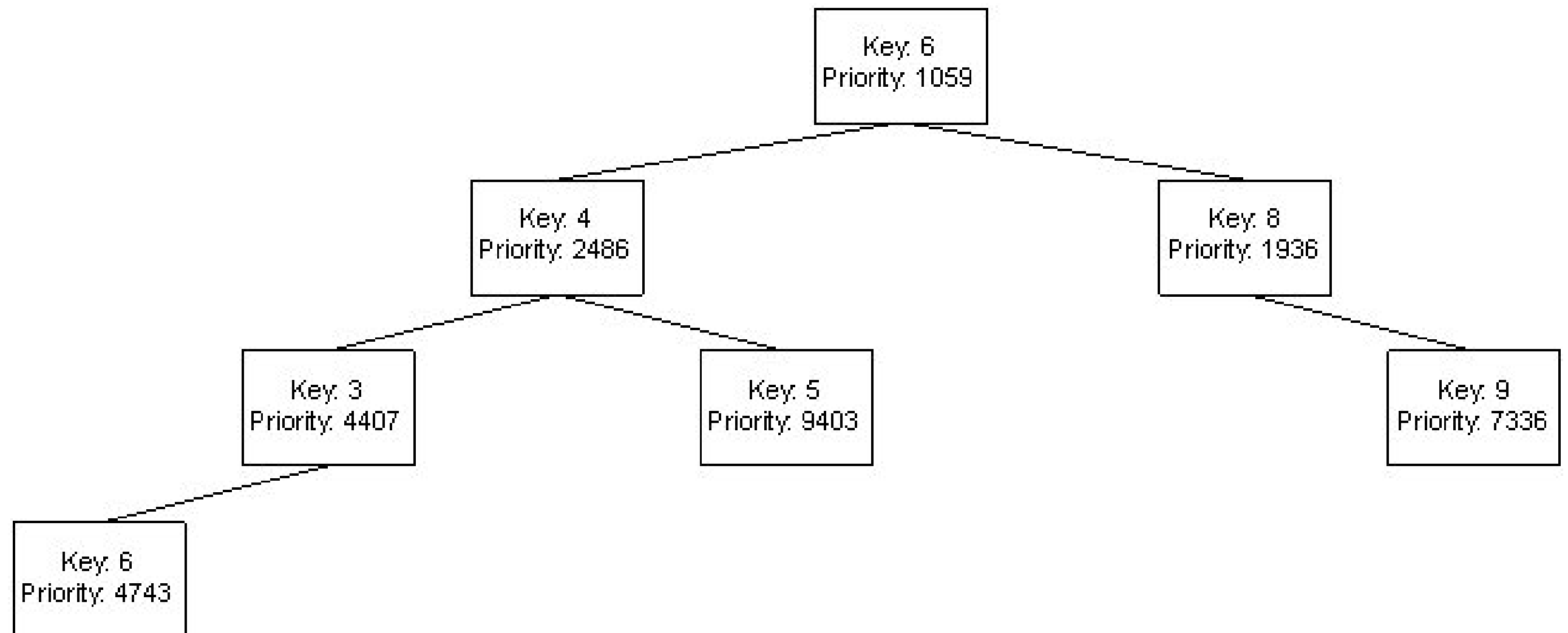
Итоговая структура является BST относительно первого значения и кучей относительно второго.



Декартово дерево



Полезная статья про
реализацию
Декартового дерева





Спасибо!

