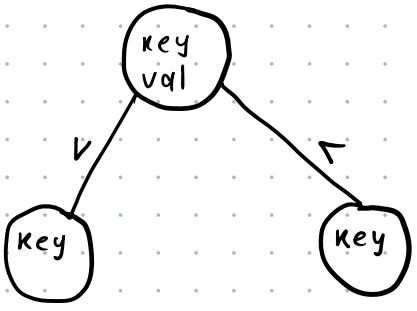
20.02.26

ДЕРЕВЬЯ ПОИСКА

[(key1, val1), (key2, val2), ...]

ПОИСК В МАСС. O(n)

) МАССИВЫ

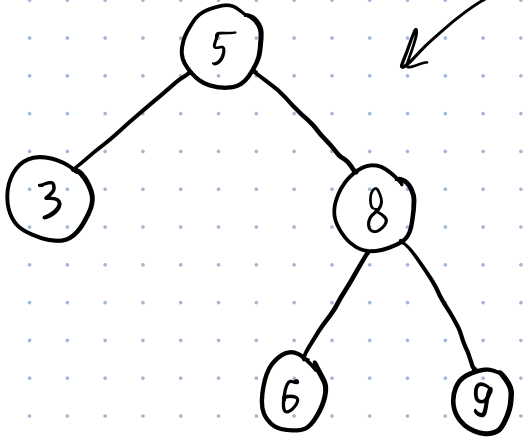


- ДОБАВЛ.
- ПОИСК
- УДАЛ.

) O(h)
O(log n)

КЛЮЧИ РАЗНЫЕ!

ПРИМЕР ДЕРЕВА



В ПРАВОМ ПОДД. ВСЕ КЛЮЧИ БОЛЬШЕ КЛ. КОРНЯ

ФУНКЦИЯ ПОИСКА:

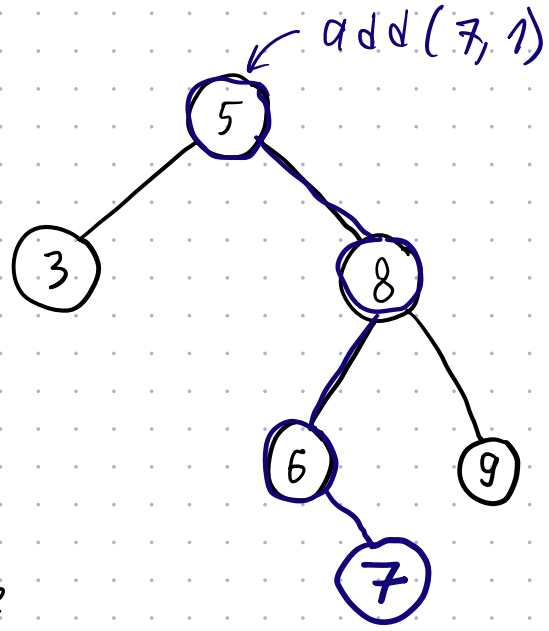
```
def find(key):
    curr = self.root

    while (curr is not None):
        if (curr.key == key):
            return curr.value
        elif (curr.key > key):
```

```
class Node
    self.r = ....

class BST
    Binary search tree
```

```
            curr = curr.l
        else:
            curr = curr.r
    return None
```

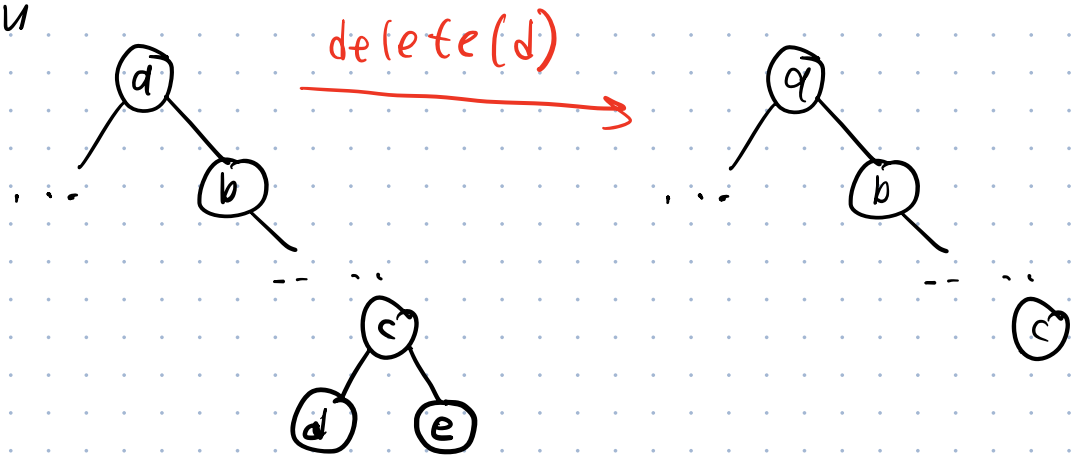
```
class BST:
    __init__(self):
        self.root = None

    def find(self, key):
        ......

    def add(self, key, val):
    ..... ..... ....
```

```
def add(self, key, val):
    new_node = Node(key, val)
    if (self.root is None):
        self.root = new_node
        return

    curr = self.root

    while (True):
        if (curr.key > key):
            if(curr.l is None):
                curr.l = new_node
                return
            else:
                curr = curr.l
        elif (curr.key < key):
            if(curr.r is None):
                curr.r = new_node
                return
            else:
                curr = curr.r
        else:
            print("ignoring .....")
```

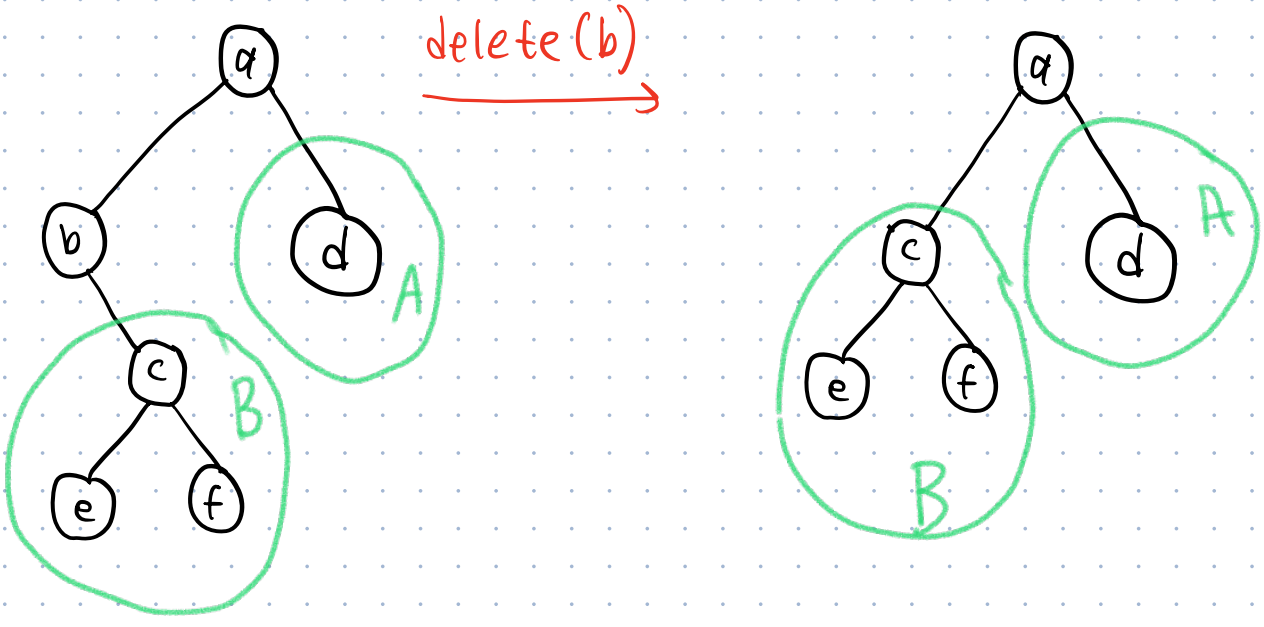


delete

0 ДЕТЕЙ



1 РЕБЁНОК

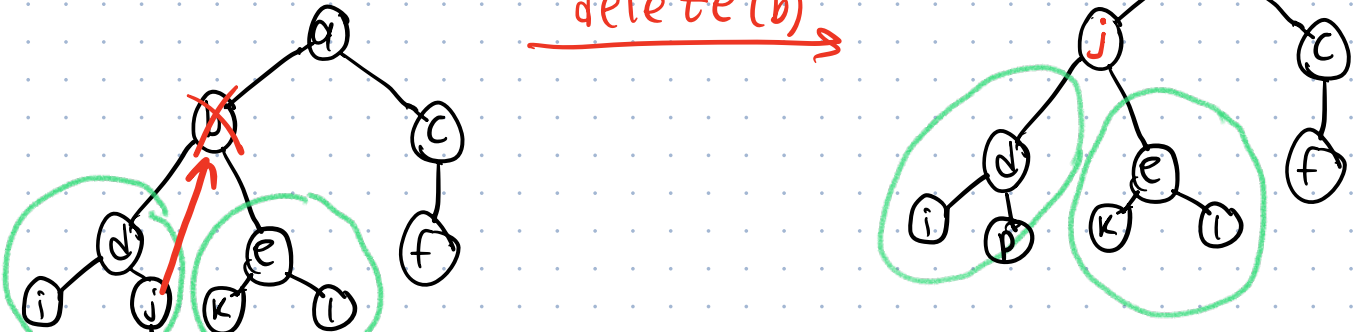


a ">" B

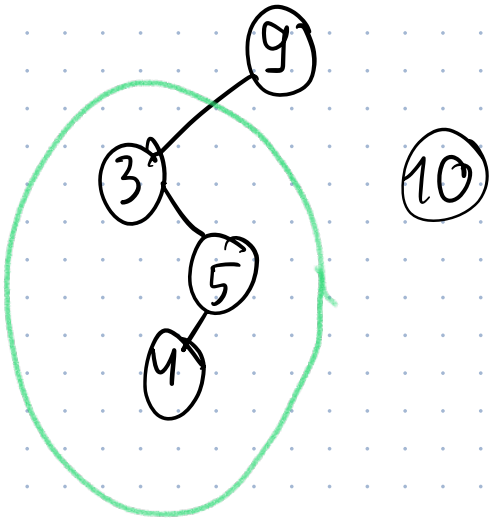
a "<" A

=> РЕЗУЛЬТАТ ТАКЖЕ ЯВЛ. ДЕР. ПОИСКА

2 РЕБЁНКА



КАК НАЙТИ max?



10

ИДЁМ ВПРАВО, ПОКА НЕ ОБН. None