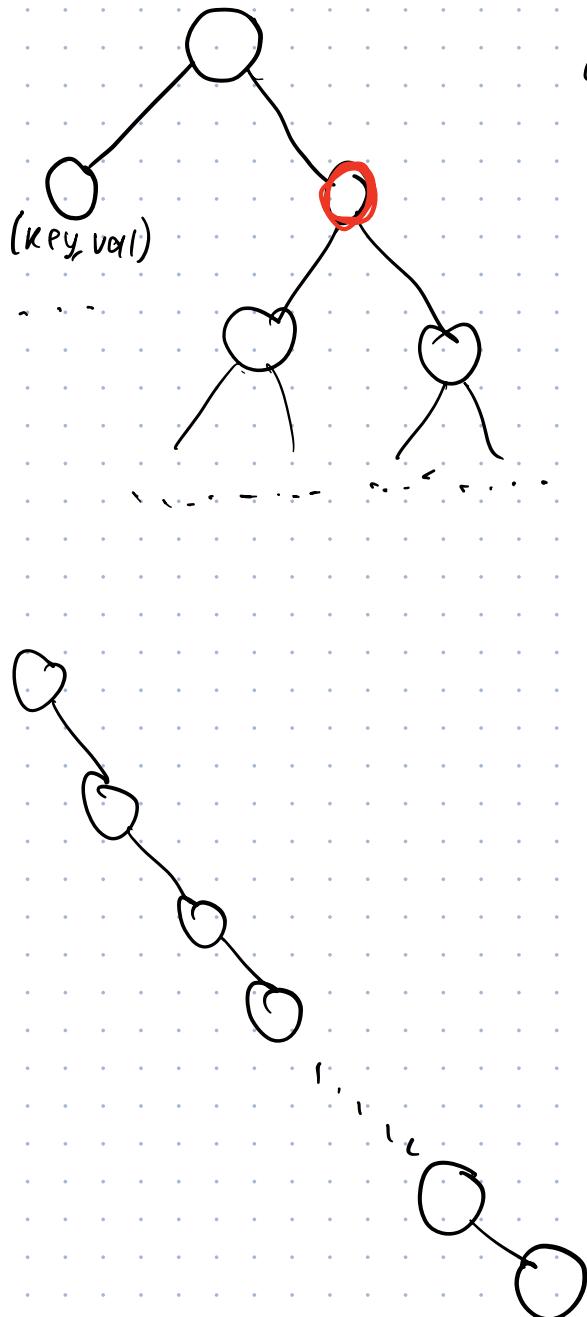


19.02.26

## ДЕРЕВЬЯ II

КАК НАЙТИ ЭЛЕМ? class BST:

```
def find(key):
    curr = self.root
    while (curr is not None):
        if (curr.key == key):
            return curr.val
        elif (curr.key < key):
            curr = curr.r
        else:
            curr = curr.l
    return None
```



## УДАЛЕНИЕ ЭЛ. ИЗ ДЕРЕВА ПОИСКА

НА ВХОДЕ:

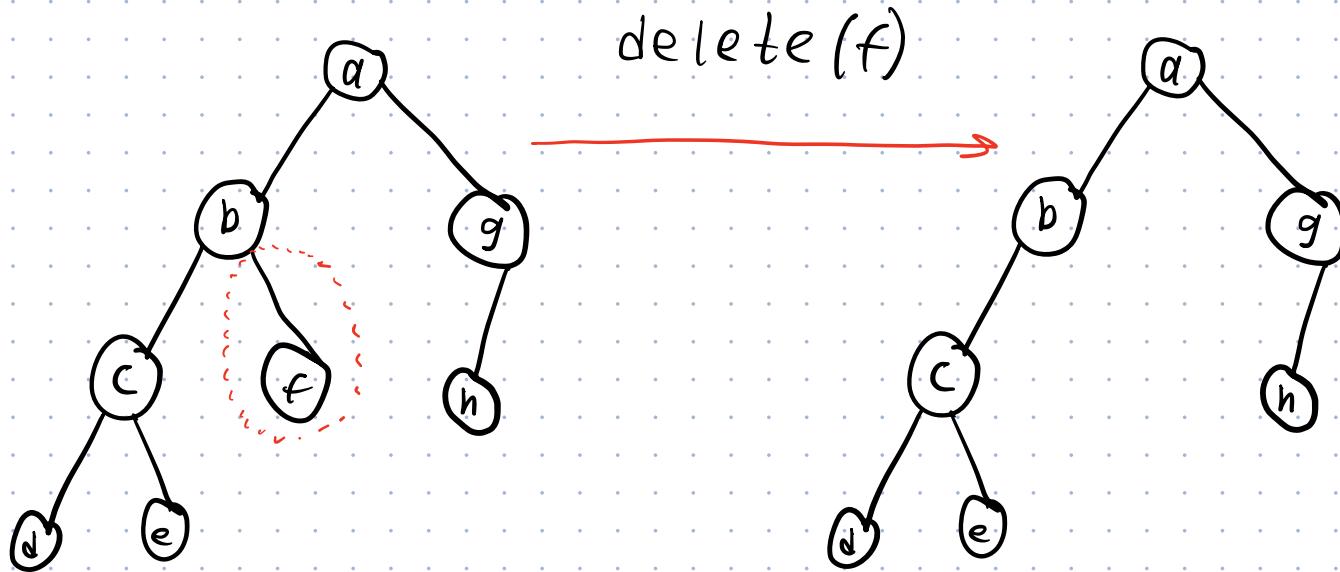
- ДЕР. ПОИСКА С УН. КЛ.
- КЛЮЧ, ПО КОТ. УДАЛЕМ

НА ВЫХОДЕ:

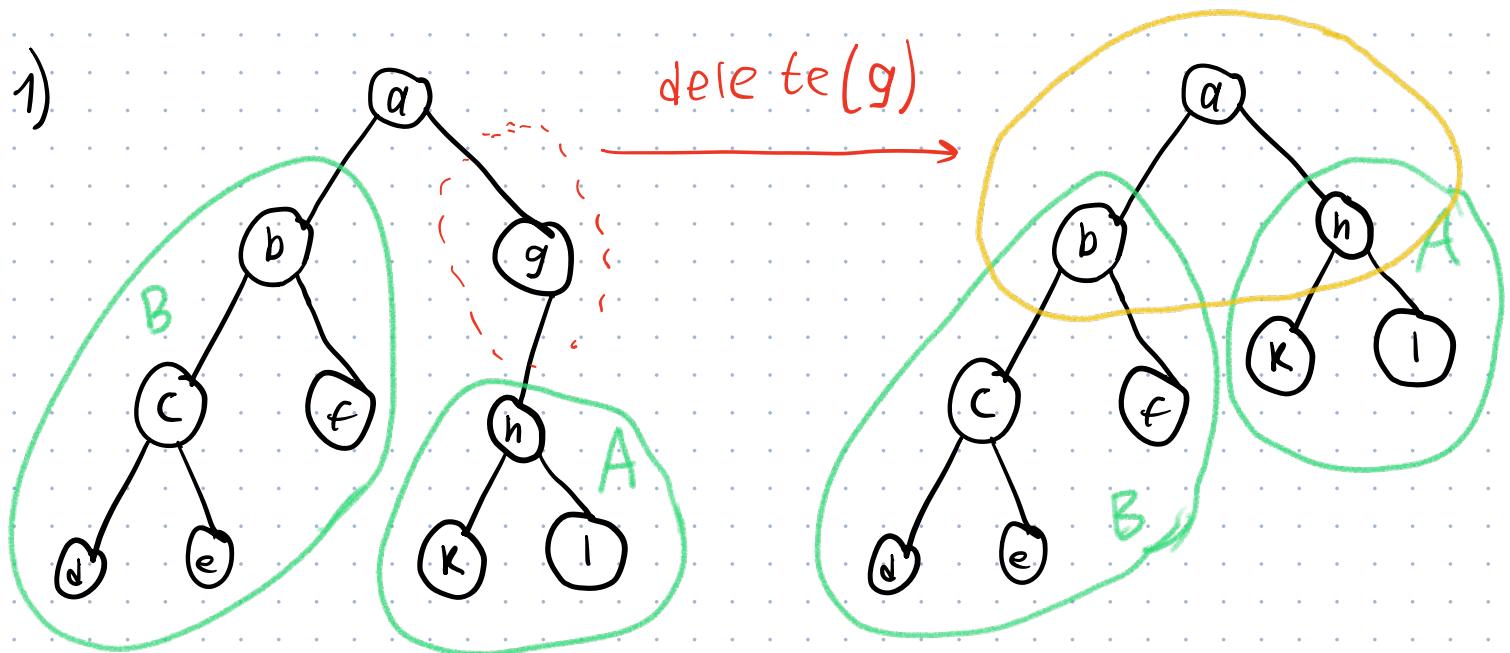
- ДЕРЕВО БЕЗ ЭЛЕМ.
- ЭЛЕМЕНТ

3 ВАРИАНТА: 0 ДЕТ., 1 РДБ., 2 РДБ.

0)



1)



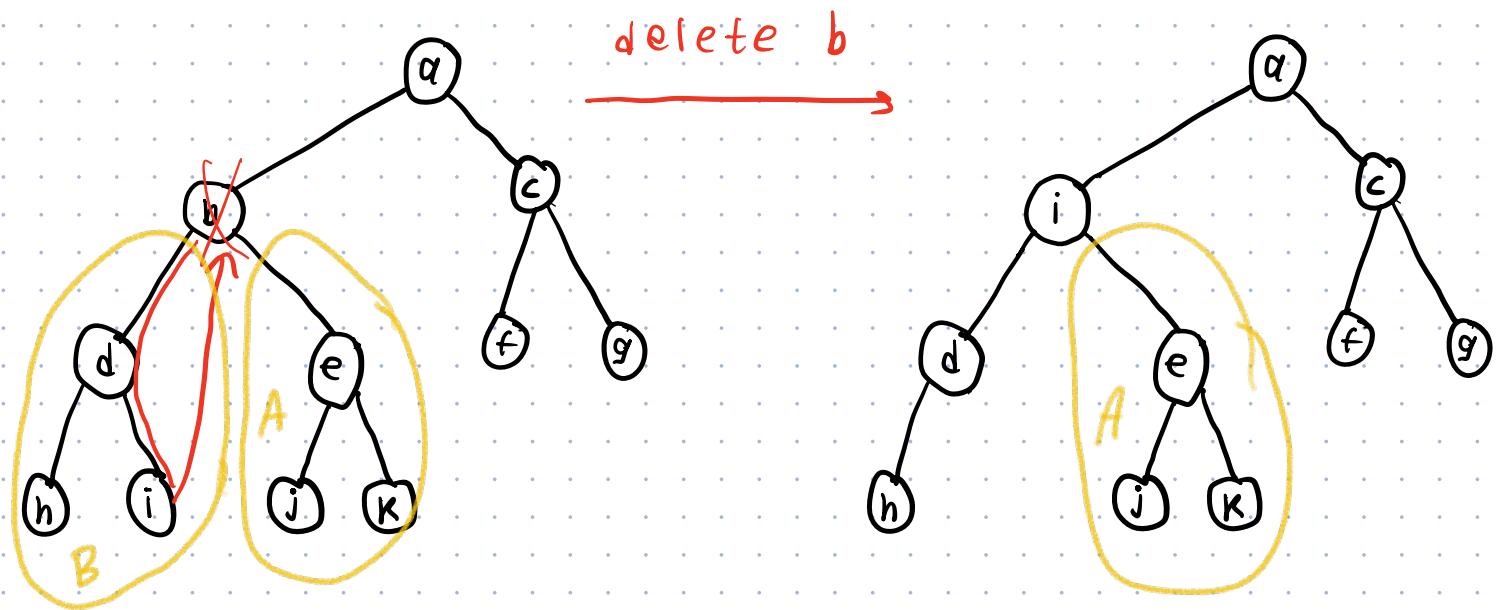
ПОКАЖЕМ КОРРЕКТНОСТЬ

$X > Y : \forall x \in A, x > \forall y \in B, y$

- 1) ВНУТРИ A, B ОП. НЕ БЫЛО
- 2)  $a.key > x.key \wedge x \in B$
- 3)  $a.key < y.key \wedge y \in a.r \Rightarrow a.key < c.key \wedge c \in A$

СЛЕД., СВ-ВО ДЕРЕВА СОХР.

## 2) 2 РЕБЕНКА



АЛГ. ПОИСКА МАХ:

ИДТИ НАПРАВО, ПОХА

НЕ ДОСТИГИ. NODE.R == None

ОБОСНИУЕМ КОРРЕКТНОСТЬ

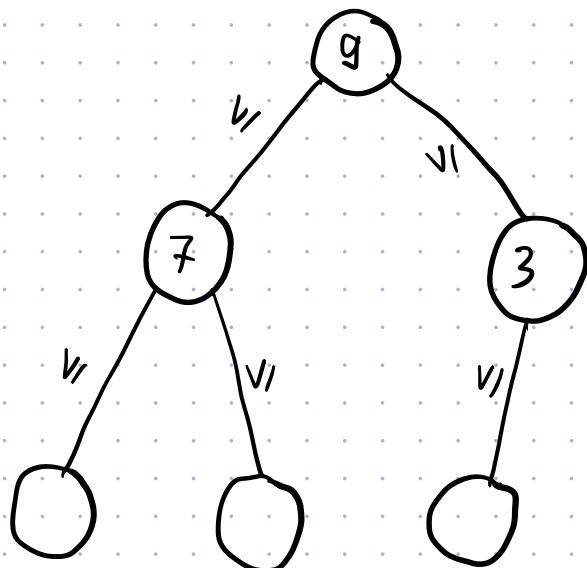
- 1)  $b.\text{key} < x.\text{key} \quad \forall x \in A$
  - 2)  $b.\text{key} > x.\text{key} \quad \forall x \in B$
  - 3)  $B.\text{max} \geq x.\text{key} \quad \forall x \in B$
  - 4)  $\exists m = B.\text{max}; \text{ ТОГДА}$   
 $m.\text{key} < b.\text{key} < a.\text{key} \quad \forall a \in A$
- $m.\text{key} \geq x.\text{key} \quad \forall x \in B$

СЛЕД.,  $m$  МОЖЕТ БЫТЬ НОВЫМ КОРНЕМ ДОД., КОРНЕМ  
КОД. БЫЛ  $b$

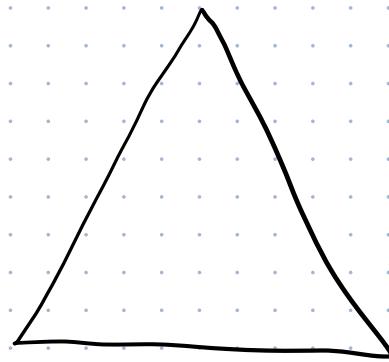
ДЛЯ УДАЛЕНИЯ  $m$  ВЫЗОВЕМ  $\text{delete}(m)$ ; ТАМ БУДЕТ ОДИН  
1 СЛУЧАЙ

# КУЧА (ПИРАМИДА, НЕАР)

↙ КУЧА НА МАКСИМУМ  
(max heap)



КЛЮЧИ ПОТОМКОВ ≤ КЛ. ВЕРШ.

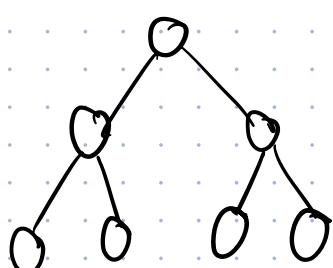


ОПЕРАЦИИ:

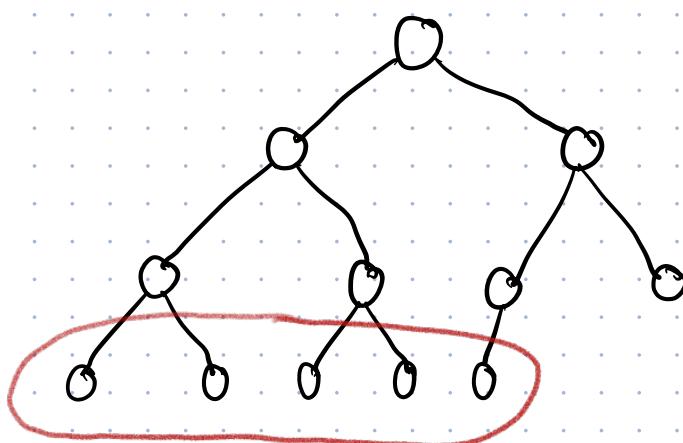
- 1) ПОСТРОЕНИЕ
- 2) ДОБАВЛ. ЭЛ.
- 3) ВЫТАСКИВАНИЕ max
- 5\*) ИЗМ. ПРИОР. = УДАЛ. + ДОБАВ.
- 4\*) УДАЛЕНИЕ

КУЧА - ПОЧТИ ПОЛНОЕ БИНАРИ. ДЕРЕВО

ПОЛНОЕ:

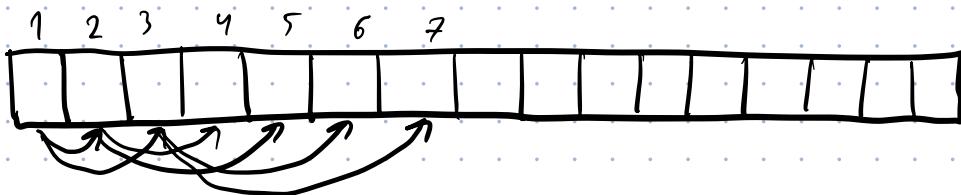


ПОЧТИ ПОЛНОЕ:

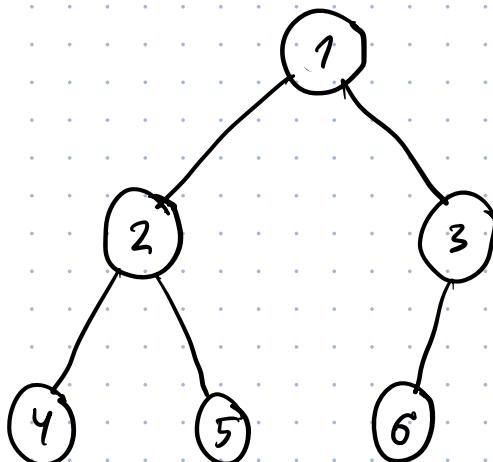
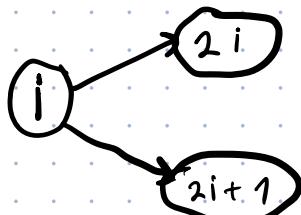


ЗАПОЛНЕНЫ ВСЕ СЛОУ, КРОМЕ  
(ВОЗМОЖНО) ПОСЛЕДНЕГО; ОН  
ЗАПОЛНЯЕТСЯ СЛЕВА НА ПРАВО

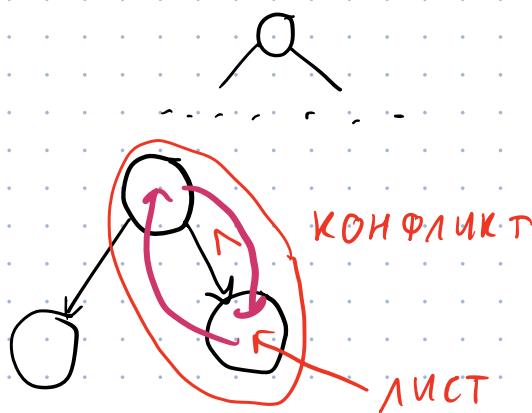
# КАК ХРАНИТЬ КУЧУ В МАССИВЕ?



СВЯЗЬ НОМЕРОВ РОД. И ДЕТЕЙ:

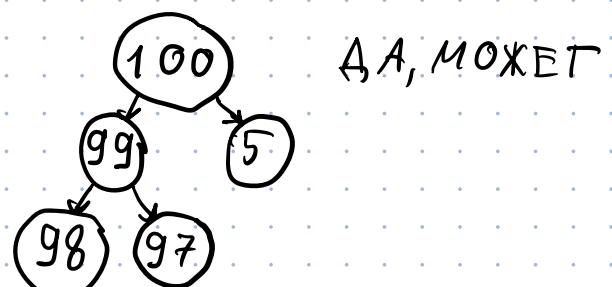


Добавл. эл.; восст. св-ва кучи



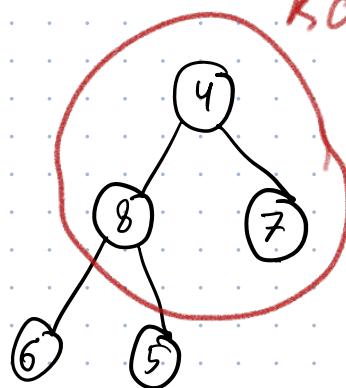
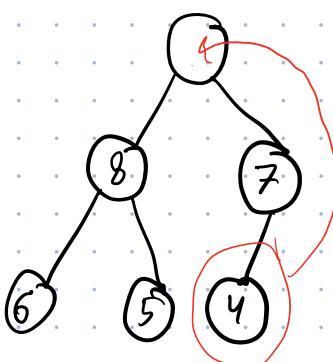
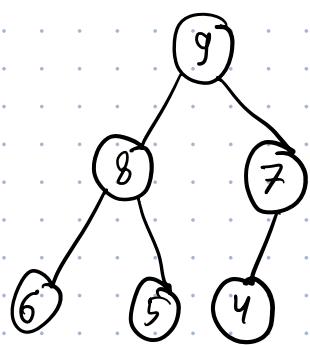
А КСТАТИ!

МОЖЕТ ЛИ ЭЛЕМ. С  
БОЛЕЕ НИЗКОГО СЛОЯ  
БЫТЬ БОЛЬШЕ ЭЛ. С  
БОЛЕЕ ВЫСОКОГО?

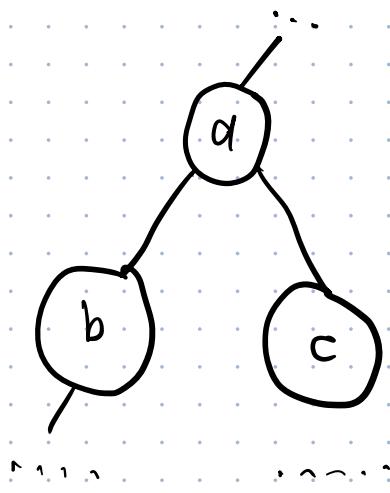


ДА, МОЖЕТ

# УДАЛЕНИЕ ЭЛ. С МАКС ПРИОРИТЕТОМ



КОНФЛИКТ

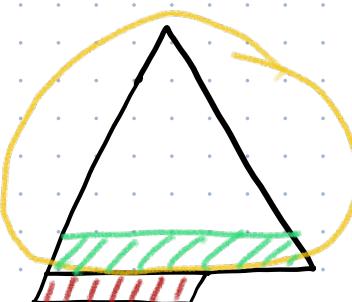
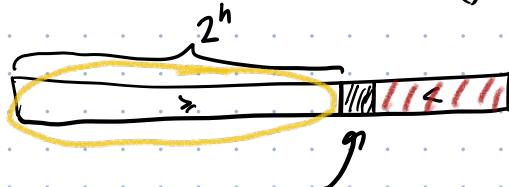


ЗА НЕ БОЛЕЕ ЧЕМ  $\log n$  ОБМЕНОВ СВ-ВО КУЧИ ВОССТ.

КАК ПОСТР. КУЧУ?

МОЖНО С ПОМ. СОРТИРОВКИ

МОЖНО С ПОМ. ПОРЯДК. СТАТ.:



$a_{(k)}$  — эл., кот. стоит на  $k$ -м месте в отсорт. масс.

## СОРТИРОВКА КУЧЕЙ (heap sort)

1) строим кучу

2) while (heap not empty)

$m = \text{heap}.\text{extract\_max}()$  1)  $O(1)$

2) heapify  $O(\log n)$

answer.append(m)

$O(n \log n)$