

# Деревья поиска

Задача точного поиска.

$n$  элементов, среди них найти эл-т равный  $X$ .

$O(n)$

Эл-ты можно сравнивать.

1. Статическая (набор не изменяется)
2. Динамическая (изменяется)

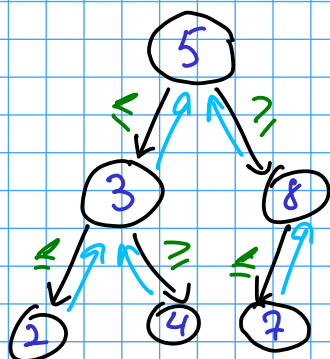
1. Отсортированный массив  
и использовать двоичный поиск.

Сложность  $O(n \log n) + O(m \cdot \log n)$   
 $m$  - # запросов.

$O(\log n)$  на запрос.

2. Деревья поиска.

- Бинарные / двоичные
- ... B-деревья



АТД:

множество:

insert (key)  
erase (key)  
find (key)

словарь:

insert (key, value)  
erase (key)  
find (key)

node:

left, right  
parent  
key

find (k, n = root):

if n = nil: return false  
if n.key > k:  
return find (k, n.left)  
else if n.key < k:  
return find (k, n.right)  
return true

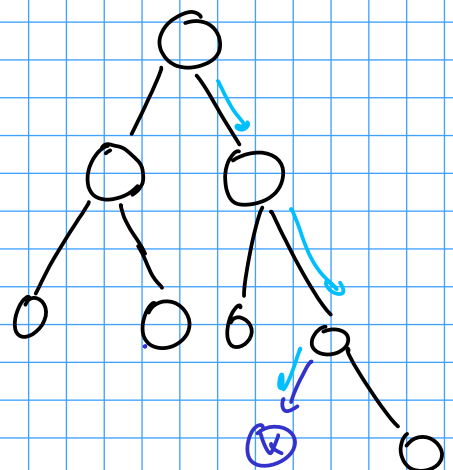
$O(h)$

insert (k, n = root)

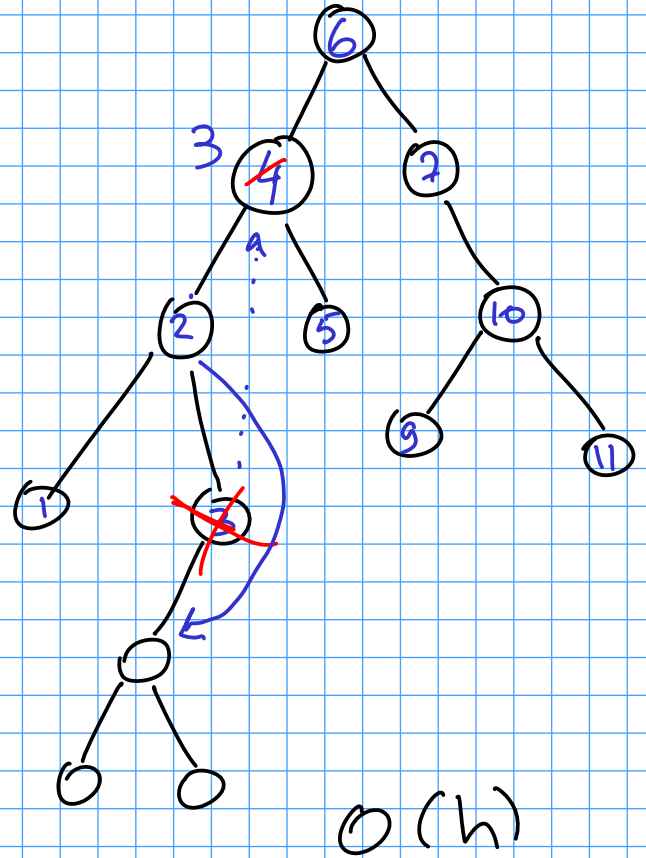
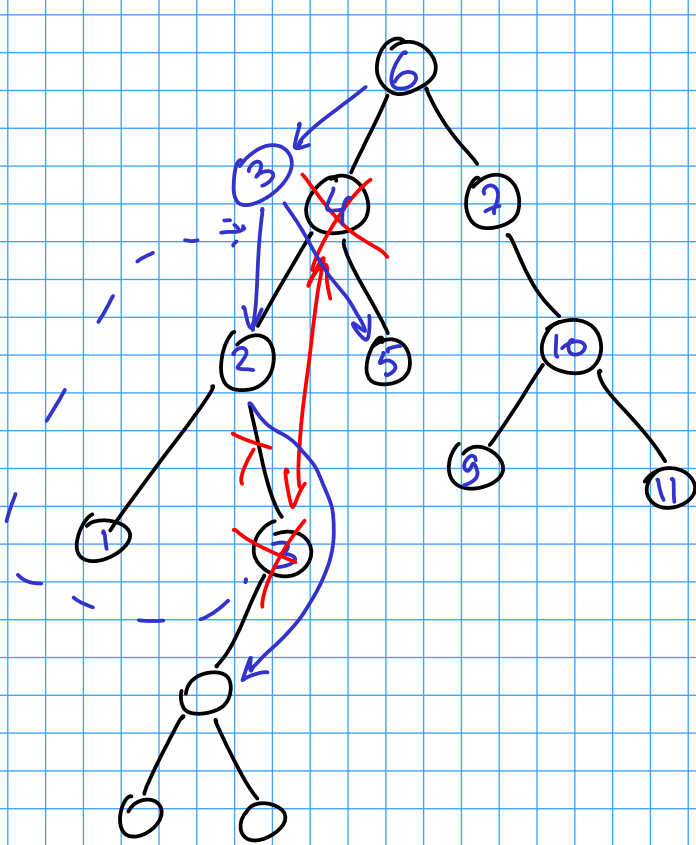
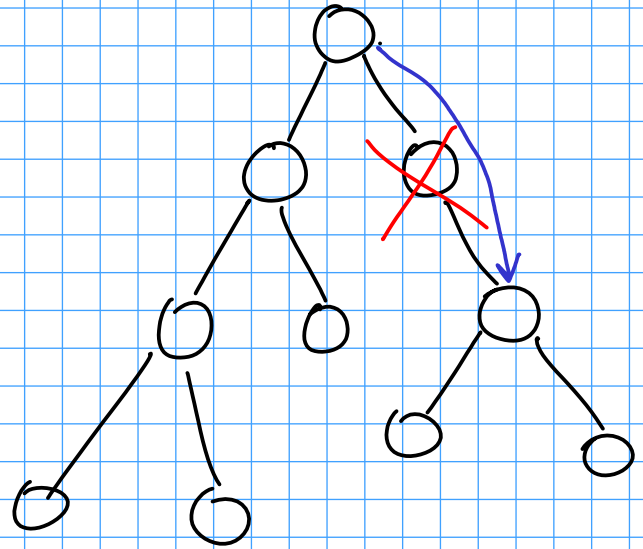
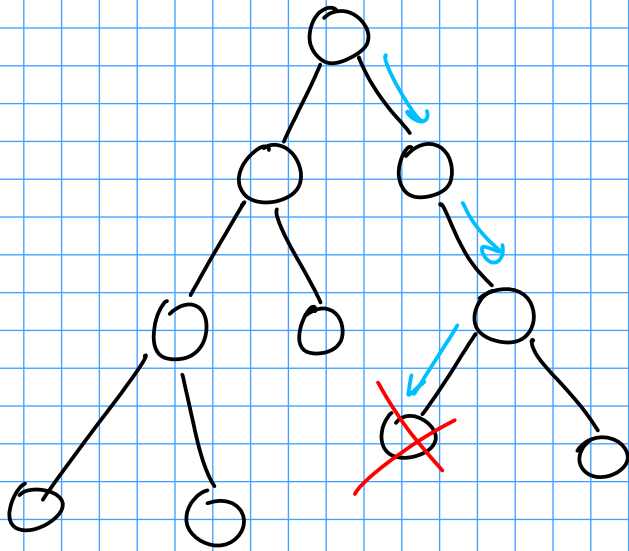
if n.key > k:  
if n.left = nil  
n.left = node (k)  
else  
insert (k, n.left)

$O(h)$

if n.key < k:  
if n.right = nil  
n.right = node (k)  
else  
insert (k, n.right)



erase(k)



$O(h)$

Простые запросы:

max()

min()

next()

prev()

$O(h)$

Сбалансированные деревья поиска:

- AVL деревья
  - Красно-черные деревья
  - ...
- }  $h = O(\log n)$

AVL # node:  $|h(\text{left}) - h(\text{right})| \leq 1$

RB # node:  $\frac{2}{3} \leq h(\text{left})/h(\text{right}) \leq \frac{3}{2}$

### AVL Деревья

$h$  — # эл-ов

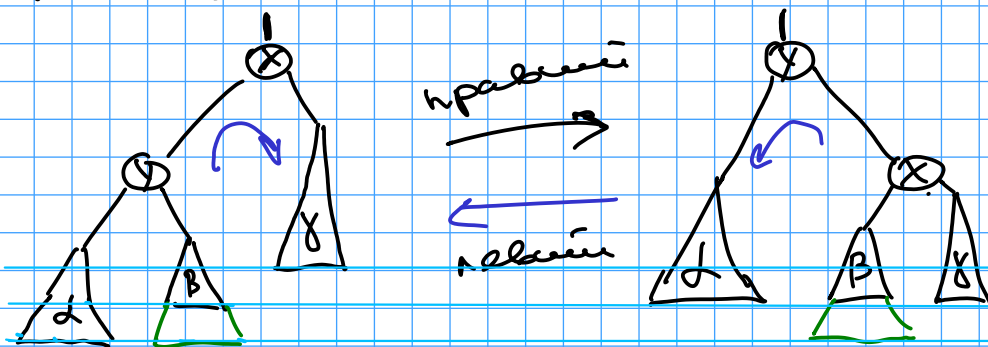
$M(h)$  — min # эл-ов в дереве высоты  $h$

$$M(h) = M(h-1) + M(h-2) + 1$$

$$M(h) \sim 1.6^h$$

Повороты:

- маленькие



- большие

