Деревья поиска

Задача точного поиска

S - множество эл-ов, $|S| = n$
x - эл-т.
Вопрос: $x \in S$? $\qquad O(n)$

] $!(a<b)$ && $!(b<a) \Rightarrow a == b$
$\underline{(a<b)} \wedge (b<a) \Rightarrow a = b$

Две постановки:
- статическая: много запросов, S - не изм.
- динамическая: много запросов, S изм.

Статическая постановка:
1. Отсортируем массив с эл-ами S $\quad O(n\log n)$
2. Будем искать бинарным поиском.
   m - # запросов

При $m = \Omega(n)$ получаем $O(\log n)$ на запрос.

Динамическая постановка.

АТД:

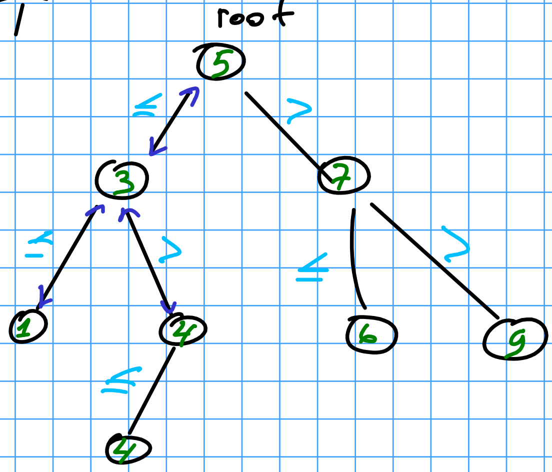
Множество:
Find (x)
Insert (x)
Delete (x)

Словарь:
Find (k)
Insert (k, v)
Delete (k)

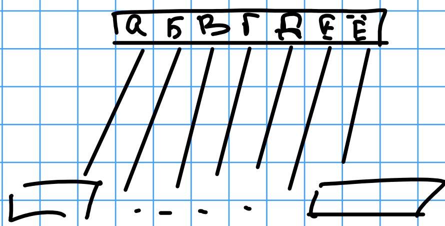
АТД множество реализуют на дереве поиска

BST

Binary Search Tree



B-дерево



Для каждого узла будем хранить left, right и parent

```
Find(x, n = root):
    if n = nil or n = x:
        return n
    if x < n:
        return Find(x, n.left)
    else
        return Find(x, n.right)
```

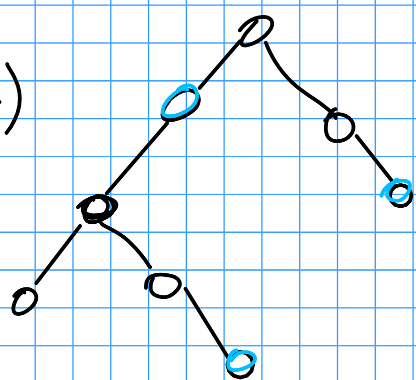
```
Tree-Minimum(n = root)
    if n = nil:
        return nil
    while n.left ≠ nil:
        n = n.left
    return n
```

Tree-Maximum(...)

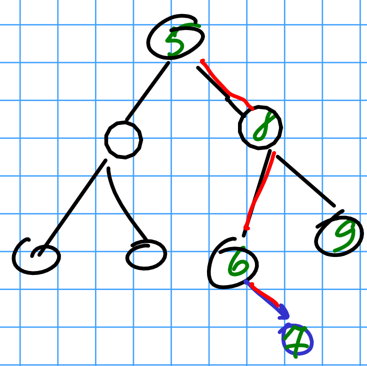
⋮

```
Successor(n):
    if n.right ≠ nil:
        return Tree-Minimum(n.right)
    while n.parent ≠ nil &
          n.parent.left ≠ n
        n = n.parent
    return n.parent
```

Predecessor(n)

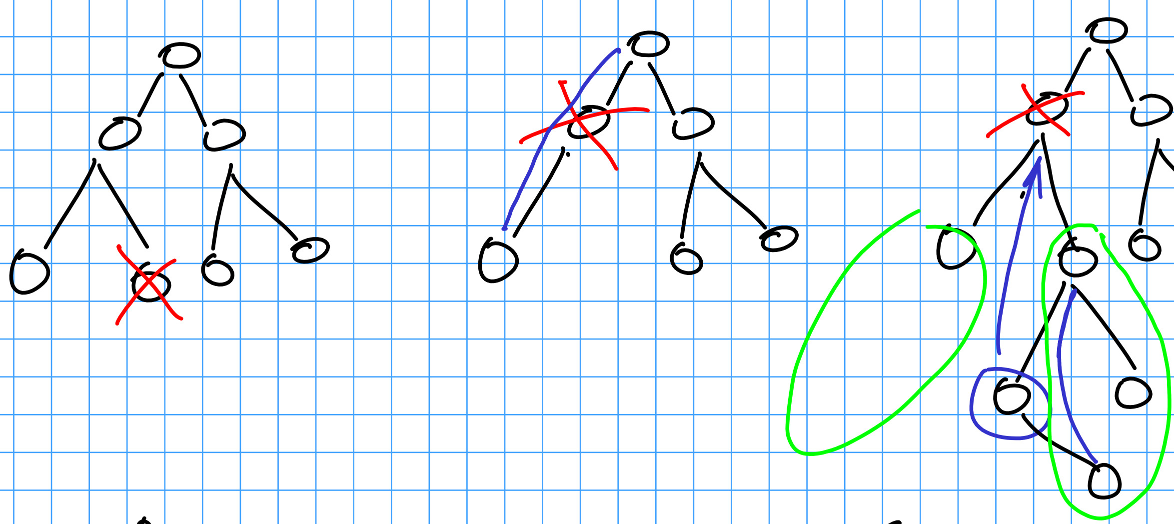


Утв: inorder обход BST даёт отсортир. набор S.



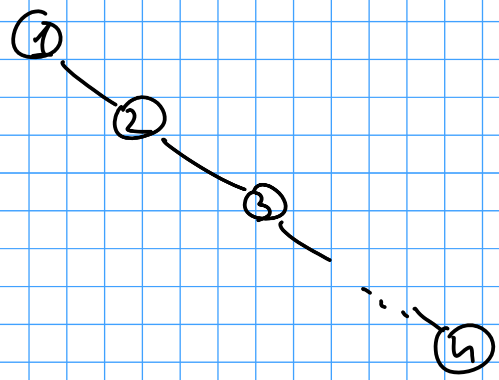
```
Insert(x)
  if root = nil
     root = x
     return
  p = root
  while true:
     if x ≤ p:
        if p.left = nil:
           p.left = x, return
        else p = p.left
     else if p.right = nil:
           p.right = x, return
        else p = p.right
```

Delete(n)



Утв: Все операции с деревом делаются за $O(h)$, h – высота дерева.

Будем стремиться к $h = O(\log n)$



$\Rightarrow \quad h = n$

Самобалансирующиеся деревья.

1. RB-Tree (см. Кормен)
2. AVL-Tree (см. Шене)
3. Splay-Tree (см. Бабенко-Левин)

AVL

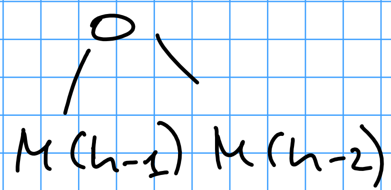
$\equiv$ $\forall$ узла: $\begin{cases} h(left) \leq h(right)+1 \\ h(right) \leq h(left)+1 \end{cases}$

$$|h(left) - h(right)| \leq 1$$

Утв: Высота AVL дерева $= O(\log n)$

$\exists$ $M(h)$ – минимальное кол-во эл-ов в AVL дереве высоты h

$M(h)$

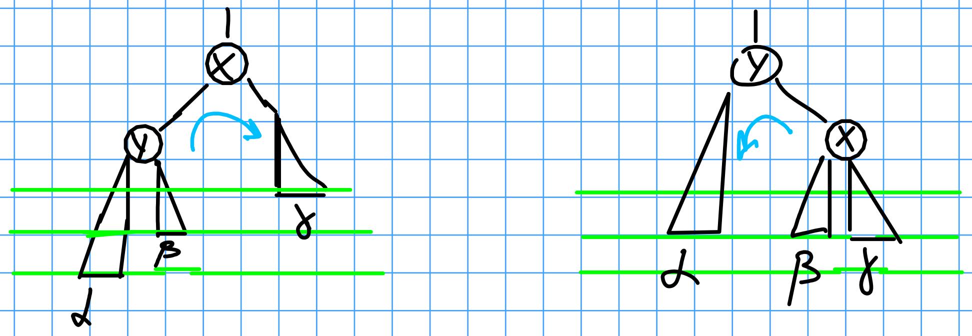


$$M(h) = M(h-1) + M(h-2) + 1$$

$$n \geq M(h) \geq Fib(h) \sim 1.6^h$$

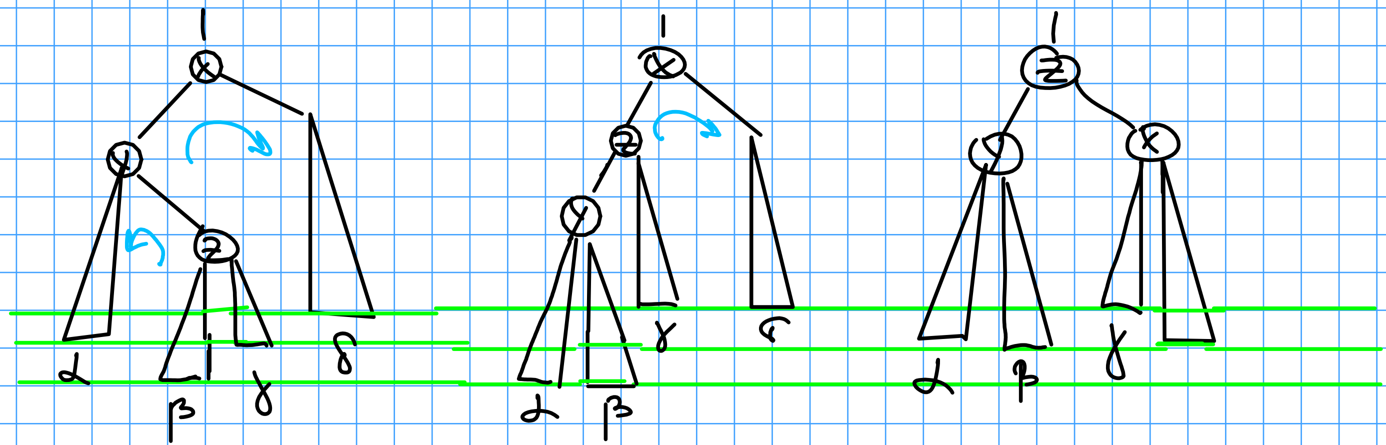
$$h = O(\log n)$$

Давайте в $\forall$ хранить высоту



Малый поворот относительно (x)–(y)

$\alpha < y < \beta < x < \gamma$ — инвариант.



Большой поворот