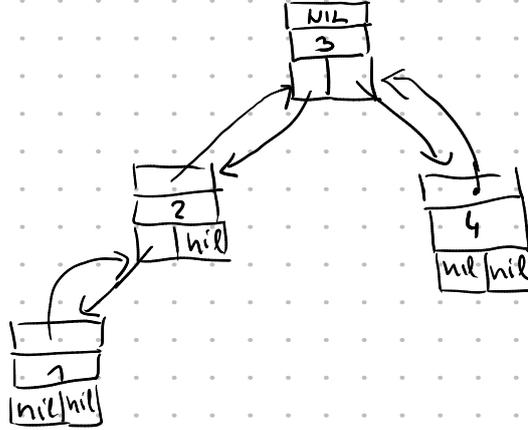
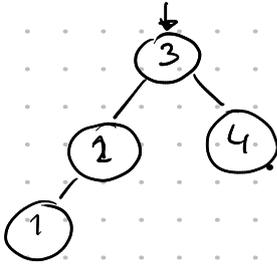


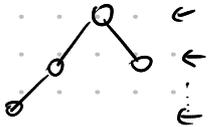
# Binární vyhledávací strom

|| Pozn: kousek poznávek na tabuli se stáhl, je nutno uvažovat zleva. Jde o seznamy: třídem' a varianty; a stromy jako spec. typ grafu.

```
struct Node
    key: Key
    left: Node
    right: Node
    parent: Node
```



{1,2,3,4}



```
Search-rec
← x: Node --- kořen stromu
← k: key
→ Node (nebo nil)
```

```
return {
    x pokud x=nil nebo x.key=k
    search-rec(x.left, k) pokud x.key > k
    search-rec(x.right, k) pokud x.key < k
```

$$\theta(1) + T(R-1)$$

$$T(R) = T(R-1) + \theta(1)$$

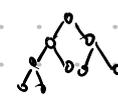
$$\Rightarrow \theta(R)$$

složitost závislá na počtu vrcholů n.

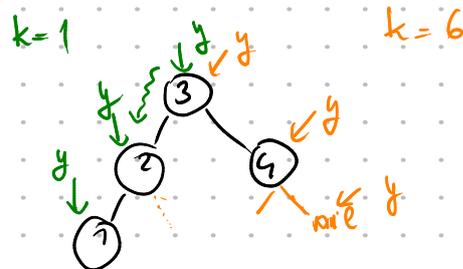
Podmínka uspořádkání: pro každé vrcholy x, y platí:

- (1) pokud je y v levém podstromu x, pak  $y.key < x.key$
- (2) pokud je y v pravém podstromu x, pak  $y.key > x.key$ .

mezi  $\theta(\lg n)$  a  $\theta(n)$



```
Search
← k: Key
← x: Node --- kořen
→ Node
```



```
y ← x
while y ≠ nil
    if y.key = k then return y
    y ← {
        y.left   y.key > k
        y.right  y.key < k
    }
return nil
```

Operace: [ min, max, příchod, poř. uvažování ]  
 [ insert, delete ]  
 [ kóda ... ]