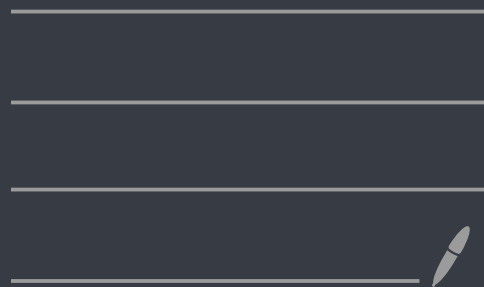


树 tree

树

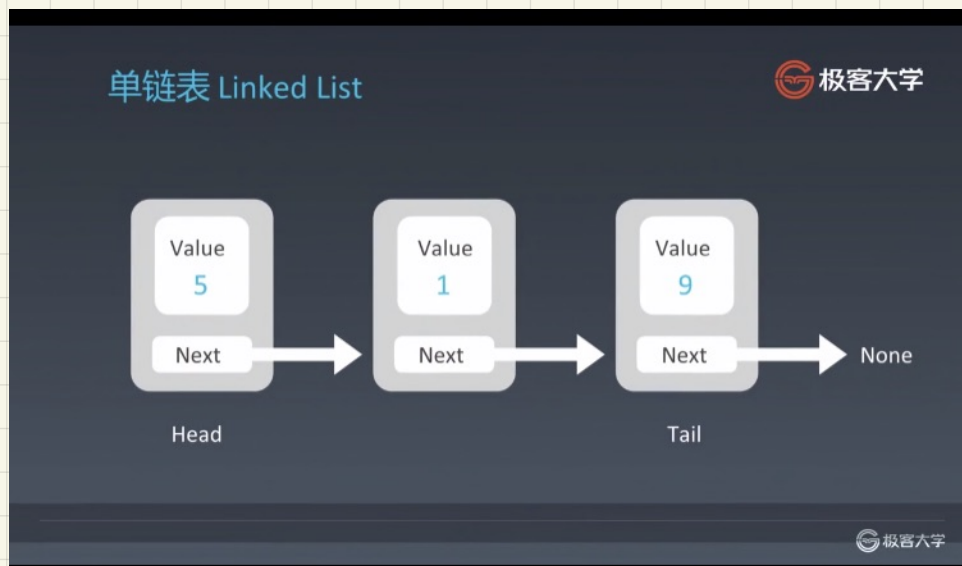
二叉树

二叉搜索树



前序：链表

- 因为链表只有后继指针（双向链表有前后继指针），所以查询时要遍历，时间复杂度是 $O(n)$ 的。
- 故出现了图表来解决（提升维度）
- 链表是特殊化的树，树是特殊化的图。
 - 只有一侧节点的树
 - 没有环的图



一. 树的基本结构

① 根节点, Root Node

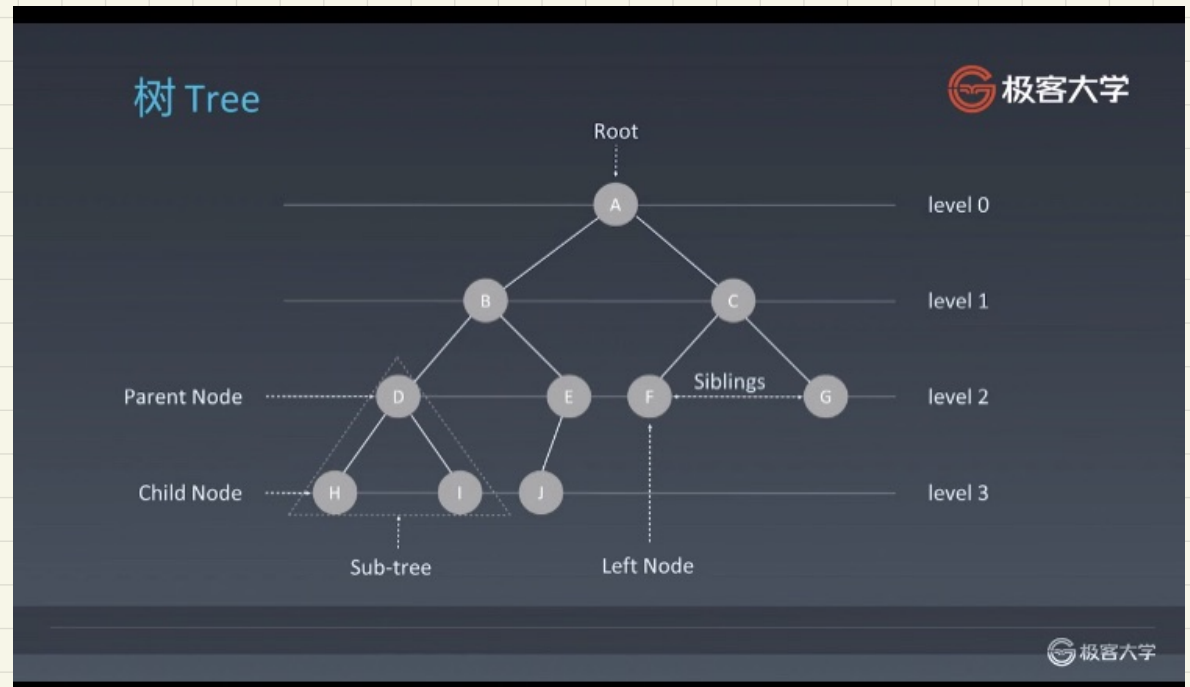
② 父亲节点, Parent Node

③ 儿子节点, child Node (Left Node, right Node)

④ 兄弟节点, siblings

树的层级 Level

子树 sub tree



二. 树的结点定义:

```
class TreeNode {  
    int val;  
    TreeNode left;  
    TreeNode right;  
    public TreeNode (int val)  
        this.val = val;  
}  
}
```

三. 树在日常中的应用

① 斐波那契数列

② 棋牌类游戏：五子棋，围棋(alphaGo)

③ 策略类游戏：星际争霸，策略树

四. 树的遍历

在树是无序的情况下, 查找树中的元素就是对树的遍历过程.

通常情况下, 树的遍历都是使用递归来完成

- 前序遍历 (pre-order): 根 → 左 → 右
- 中序遍历 (in-order): 左 → 根 → 右
- 后序遍历 (post-order): 左 → 右 → 根

五. 树的遍历代码