二叉搜索树

@M了个J

https://github.com/CoderMJLee

http://cnblogs.com/mjios

码拉松

小码哥教育 SEEMYGO

思考

- 在n个整数中搜索某个整数？（查看其是否存在）
  - 假设使用数组存放元素，从第0个位置开始遍历搜索，时间复杂度：最好 O(1)，最坏 O(n)

0	1	2	3	4	5	6	7	8	9
31	66	17	15	28	20	59	88	45	56

- 针对搜索功能，有没有改进的空间？
  - 使用二叉搜索树，最坏时间复杂度可优化至O(logn)

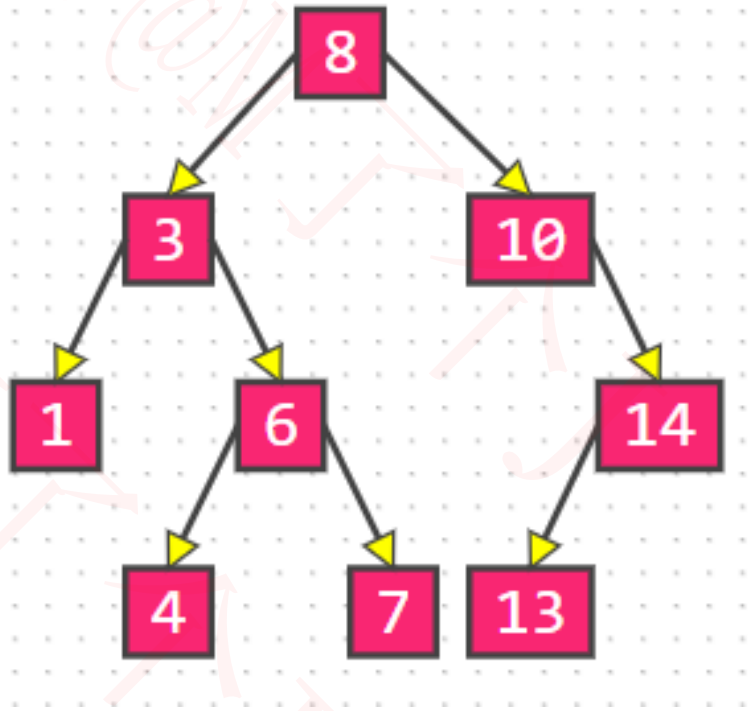
二叉搜索树（Binary Search Tree）

- 二叉搜索树是二叉树的一种，又称为：二叉查找树、二叉排序树，是应用非常广泛的一种二叉树
  - 任意一个节点的值都大于其左子树所有节点的值
  - 任意一个节点的值都小于其右子树所有节点的值
  - 它的左右子树也是一棵二叉搜索树

- 二叉搜索树可以大大提高搜索数据的效率

- 二叉搜索树存储的元素必须具备可比较性
  - 比如int、double等
  - 如果是自定义类型，需要指定比较方式

- 遇到值相等的元素该如何处理？
  - 开发者自行定义逻辑



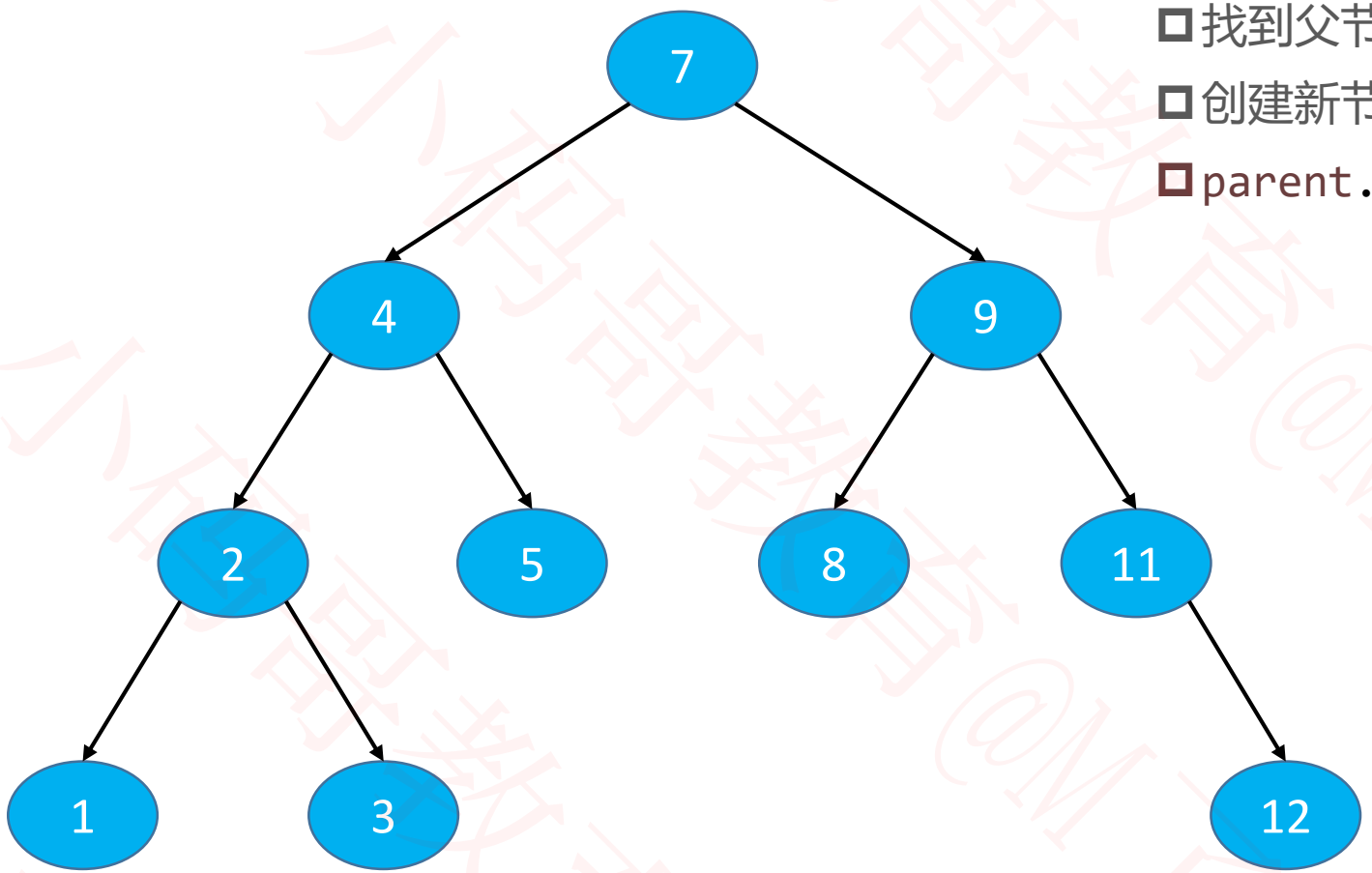
二叉搜索树的接口设计

- `int size()  // 元素的数量`
- `boolean isEmpty()  // 是否为空`
- `void clear() // 清空所有元素`
- `void add(E element) // 添加元素`
- `void remove(E element)  // 删除元素`
- `boolean contains(E element) // 是否包含某元素`

- 需要注意的是
  - 对于我们现在使用的二叉树来说，它的元素没有索引的概念
  - 为什么？

添加节点

- 比如添加12、1



- 添加步骤
  - 找到父节点parent
  - 创建新节点node
  - parent.left = node或者parent.right = node

小码哥教育 SEEMYGO

推荐一些神奇的网站

- http://520it.com/binarytrees/
- http://btv.melezinek.cz/binary-search-tree.html
- https://www.cs.usfca.edu/~galles/visualization/Algorithms.html
- https://yangez.github.io/btree-js
- https://www.codelike.in

删除节点 – 叶子节点

- 直接删除

☐ node == node.parent.left

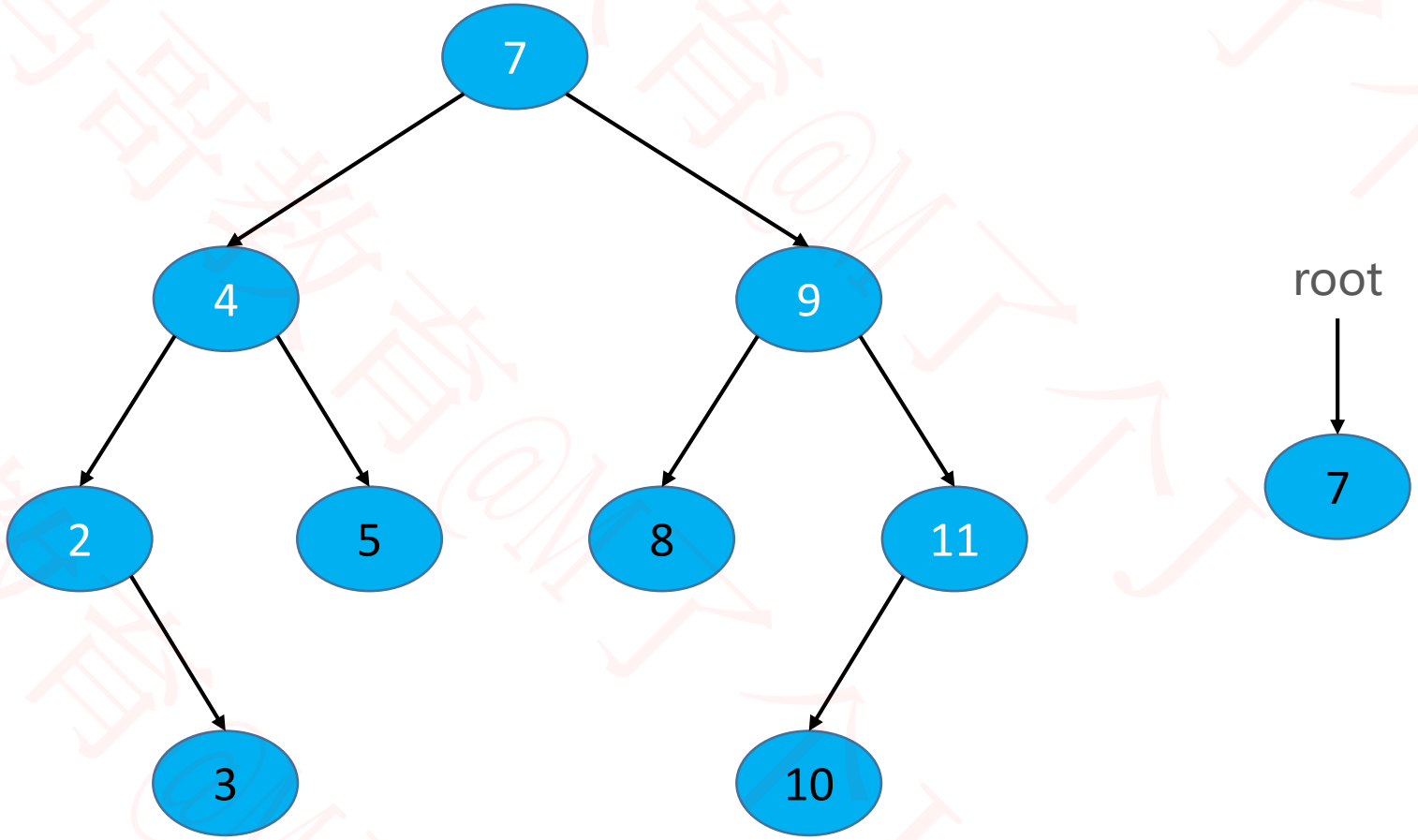
✓ node.parent.left = null

☐ node == node.parent.right

✓ node.parent.right = null

☐ node.parent == null

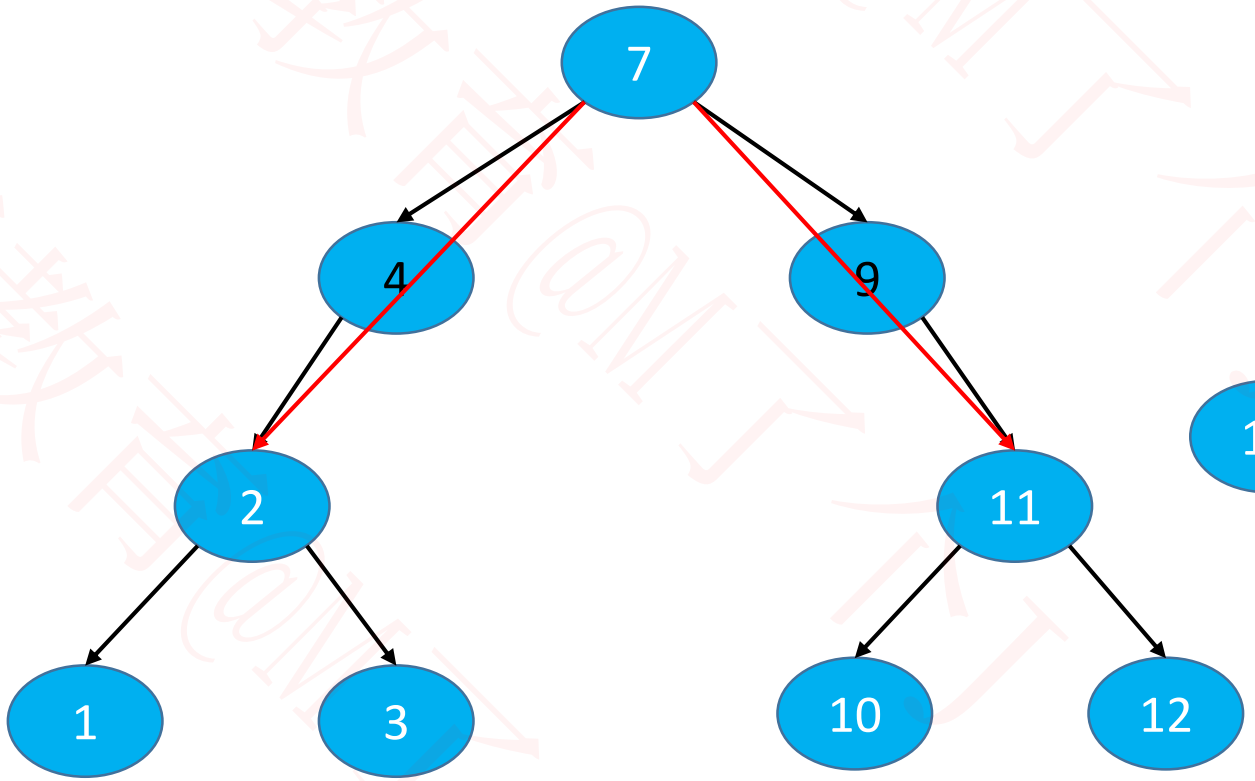
✓ root = null

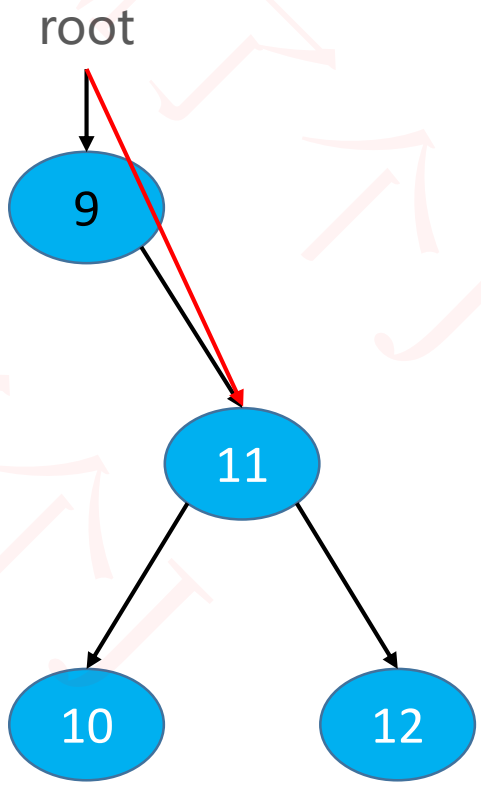


删除节点 – 度为1的节点

- 用子节点替代原节点的位置
  - child = node.left 或者 child = node.right

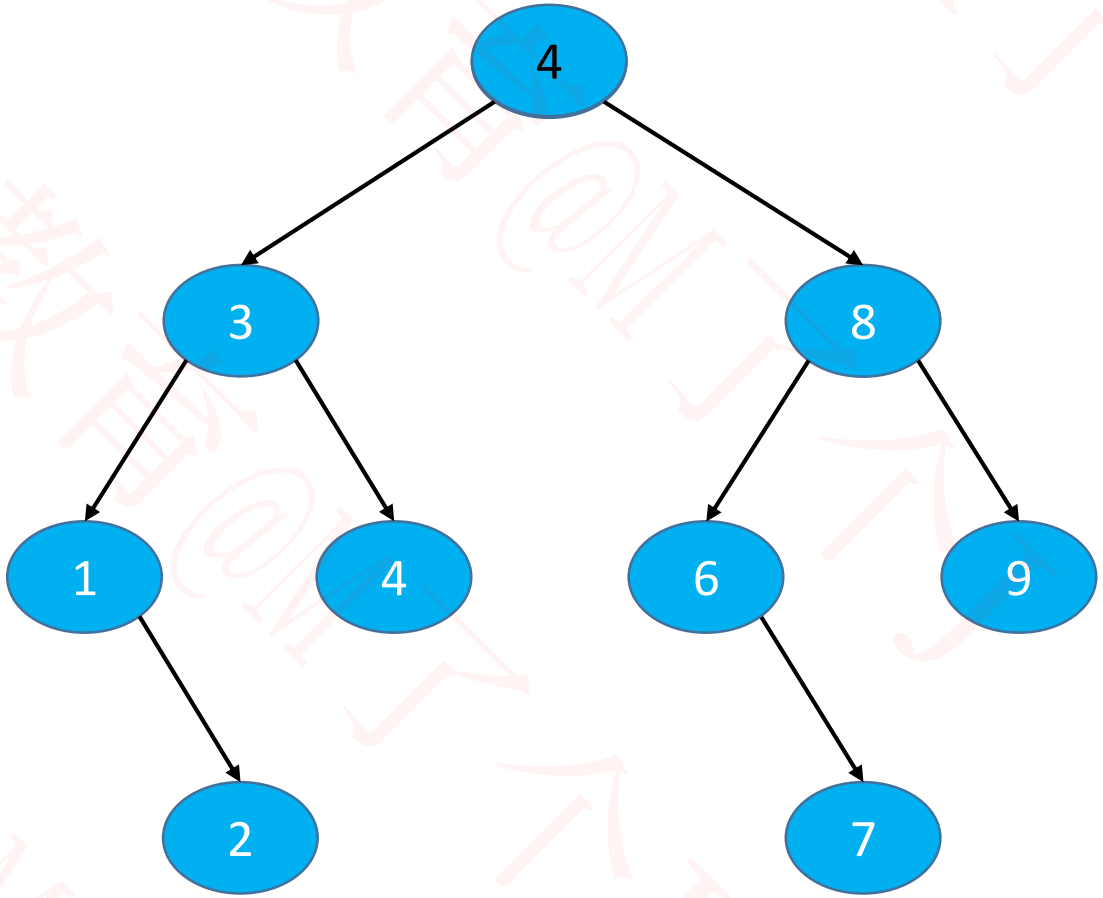
- 用child替代node的位置
  - ✓ 如果node是左子节点
    - child.parent = node.parent
    - node.parent.left = child
  - ✓ 如果node是右子节点
    - child.parent = node.parent
    - node.parent.right = child
  - ✓ 如果node是根节点
    - root = child





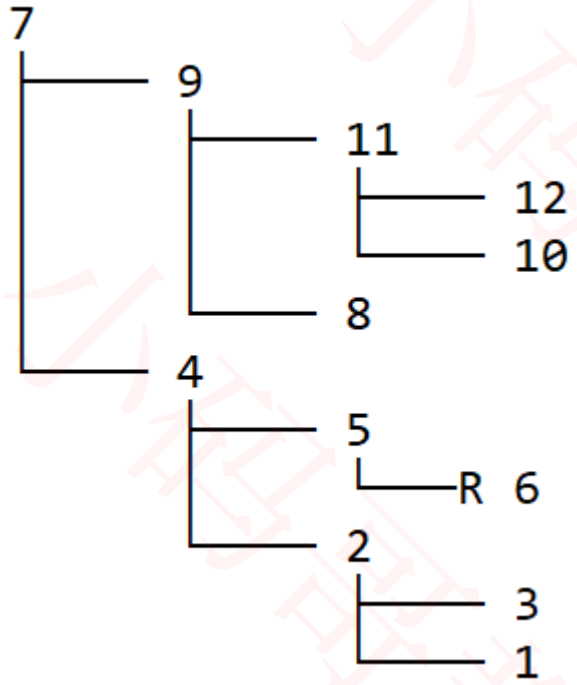
删除节点 – 度为2的节点

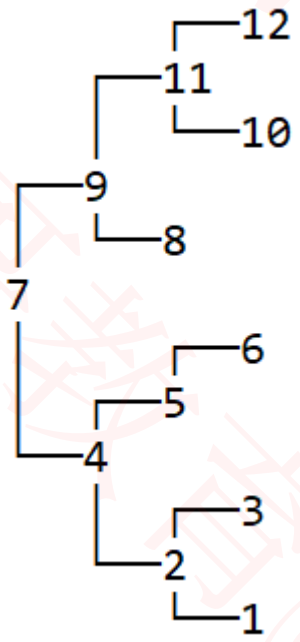
- 举例：先删除5、再删除4
- 先用前驱或者后继节点的值覆盖原节点的值
- 然后删除相应的前驱或者后继节点
- 如果一个节点的度为2，那么
  - 它的前驱、后继节点的度只可能是1和0

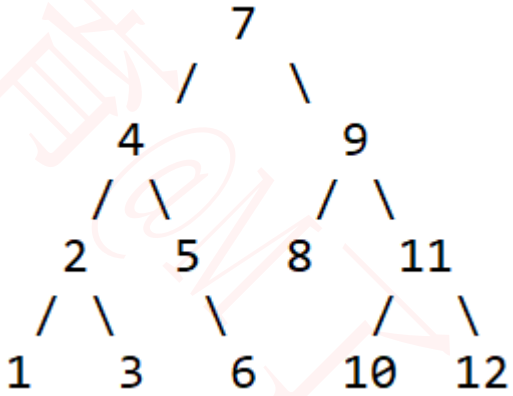


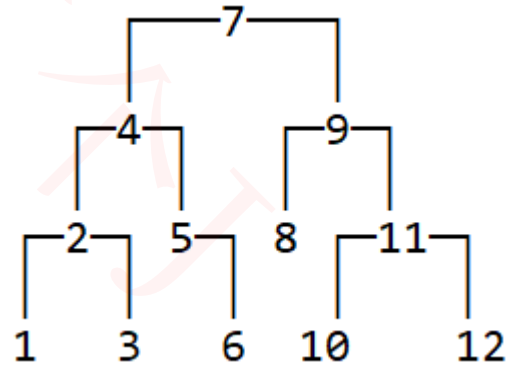
作业

- 树状形式打印二叉树
- 比如给定一个二叉搜索树：[7, 4, 9, 2, 5, 8, 11, 1, 3, 6, 10, 12]
- 尝试输出以下格式









- 开源项目：https://github.com/CoderMJLee/BinaryTrees