树

高小明
xiaomingg89@126.com

内容

- 二叉树
  - 深度优先遍历
  - 广度优先遍历
- 二叉搜索树
  - 数据结构实现
- 红黑树
- 应用：
  - 键树Trie树
  - 泛化分层

二叉树

- 层次结构
  - 家族
  - 泛化层
- 表示方法
  - XML
  - 。。。
- 二叉树
  - 节点个数

二叉树

- 数据结构

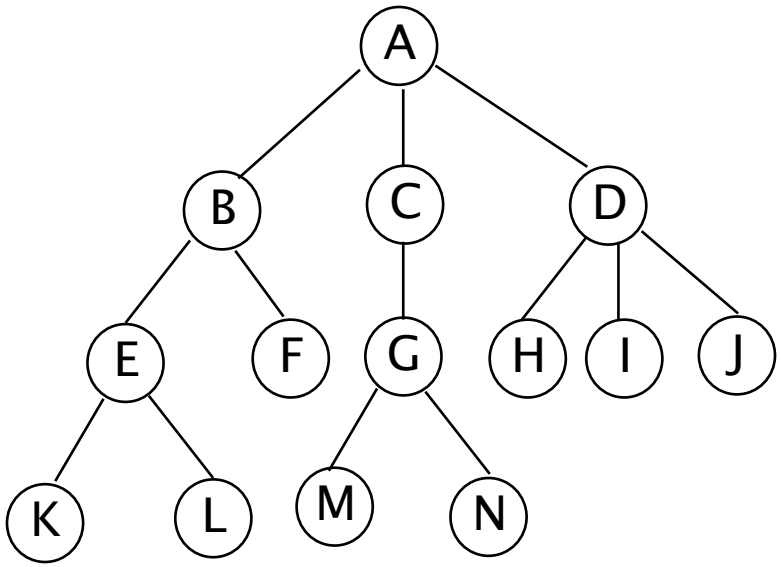
Lchild	data	Rchild

(a) 二叉链表结点

Lchild	data	parent	Rchild

(b) 三叉链表结点

- 遍历
  - 深度遍历
  - 广度遍历



二叉树的操作

- 线索化
- 求高度
- 叶子节点数

二叉树的线索化

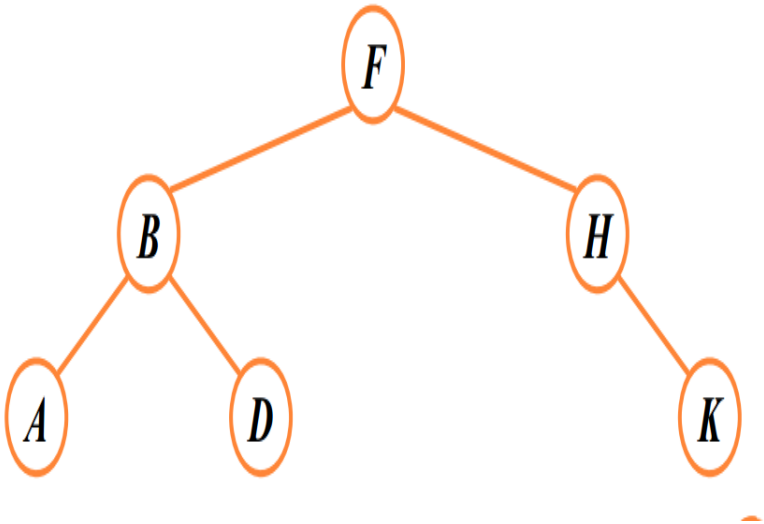
- 二叉树节点左右孩子有部分为空
- 记录后继节点
- 一般有：中序线索化

动手

- 请写出二叉树的中序线索化伪代码

二叉查找树

- 查找
- 中序遍历有序



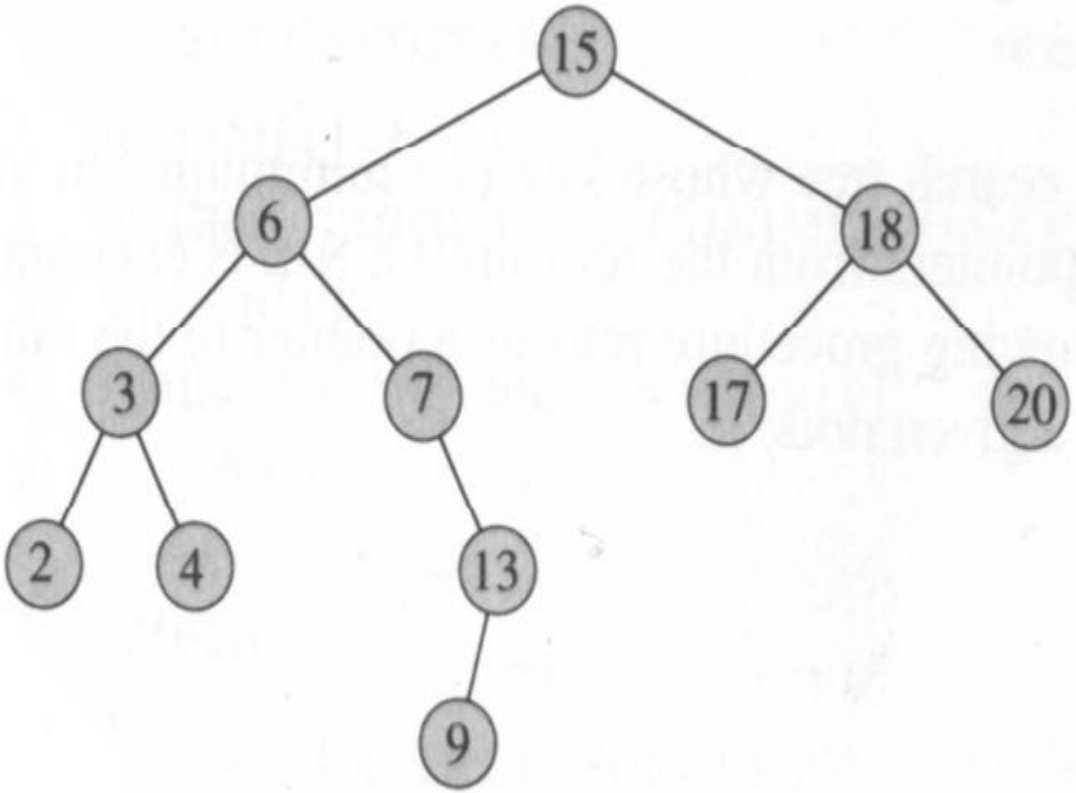
二叉查找树操作

- 建树
  - 插入

```
Void InsertTree(TreeNode *root, T elem)
 {
      if(root == NULL) root = new TreeNode(elem);
      if( elem < root->data) insert(root->left, elem)
      Else insert(root->right, elem)
}
```

删除

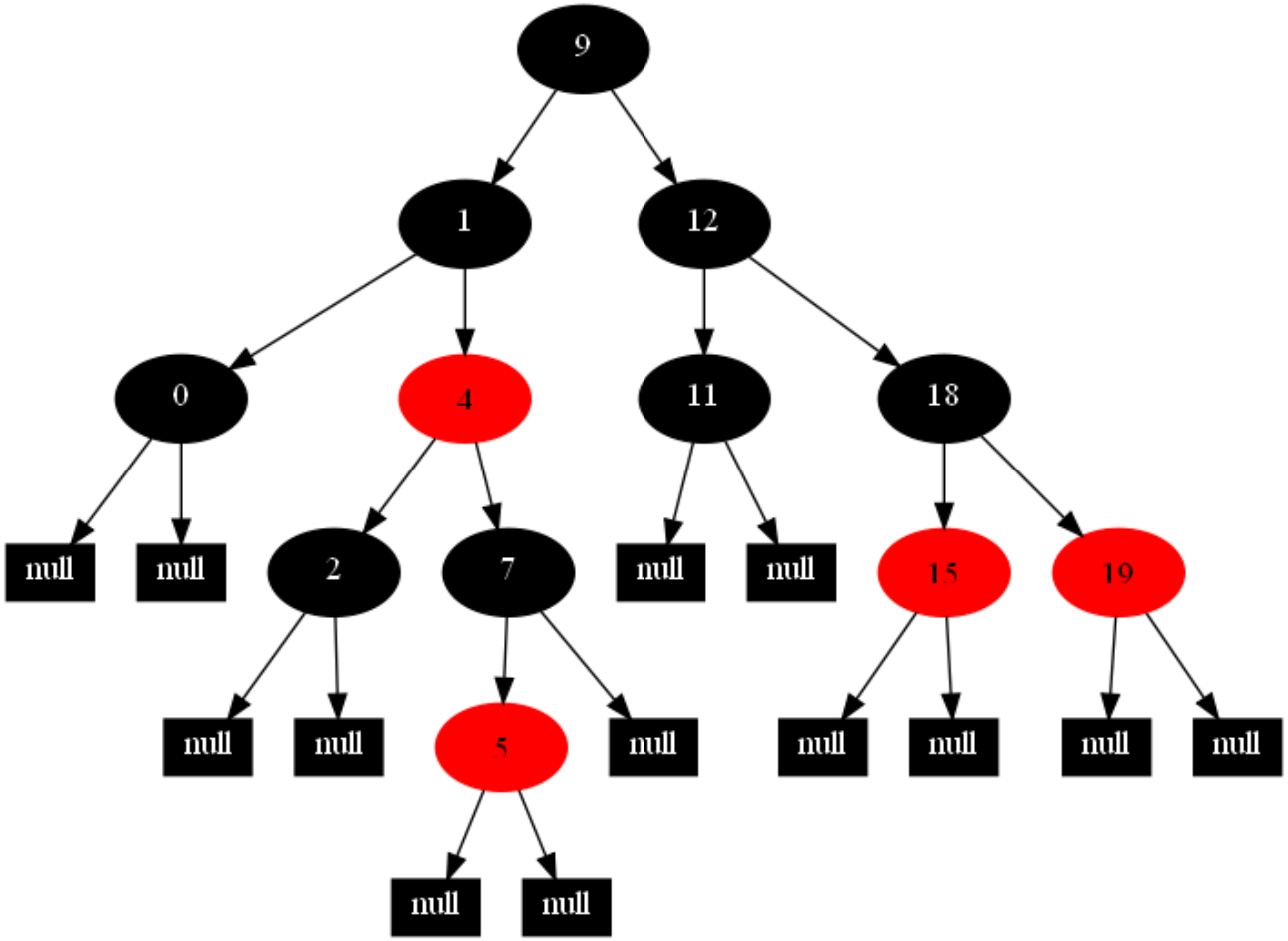
- 删除9
- 删除6
- 15？



查找

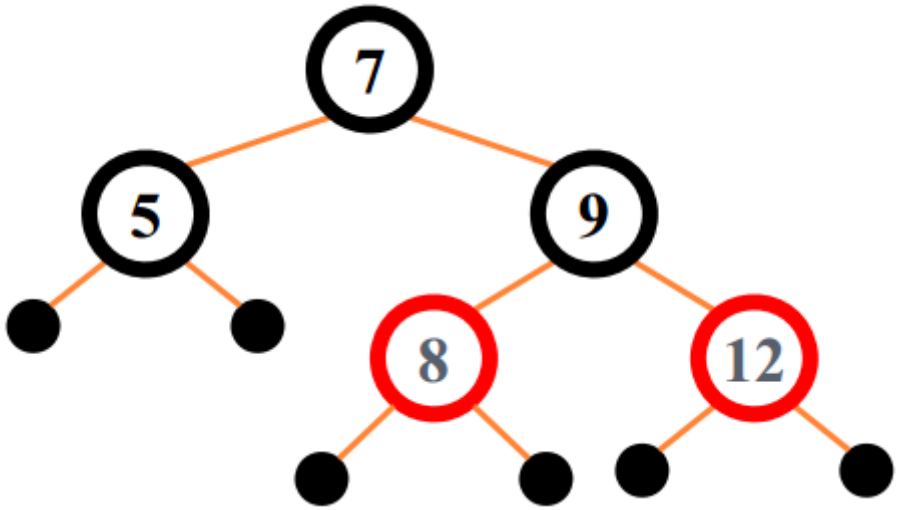
- 判断某个元素
- 查找Predecessor
  - Successor
- 查找最值
  - 最小值
  - 最大值？

2-3-4树


9
1
12
0
4
11
18
null
null
2
7
null
null
15
19
null
null
5
null
null
null
null
null
null
null
null

红黑树

- 二叉查找树不足
- 红黑树
  - 一种二叉查找树
  - 一条路径不会比其他路径长出两倍



应用场景

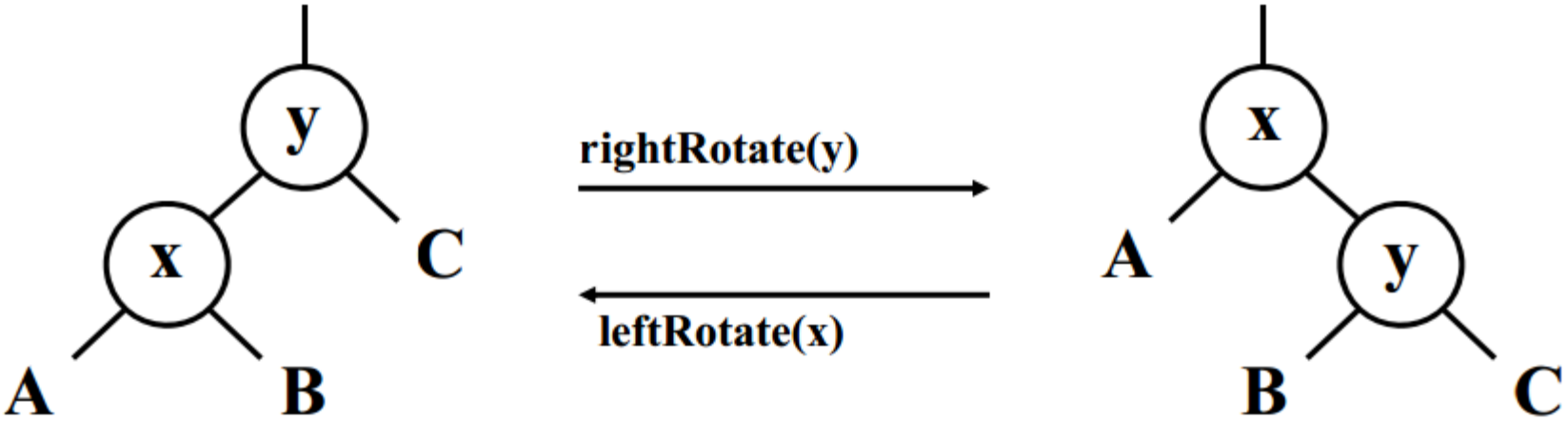
- 查找与排序
- 底层数据结构：TreeMap

红黑树的性质

- 每个节点或是红，或是黑
- 根节点是黑节点
- 每个叶节点是黑色的
- 如果一个节点是红色，则它的两个儿子是黑色
- 对每个节点，从该节点到其他子孙节点的所有路径上包含相同数目的黑节点

旋转

- 维护红黑树
- 修改指针
- 时间复杂度

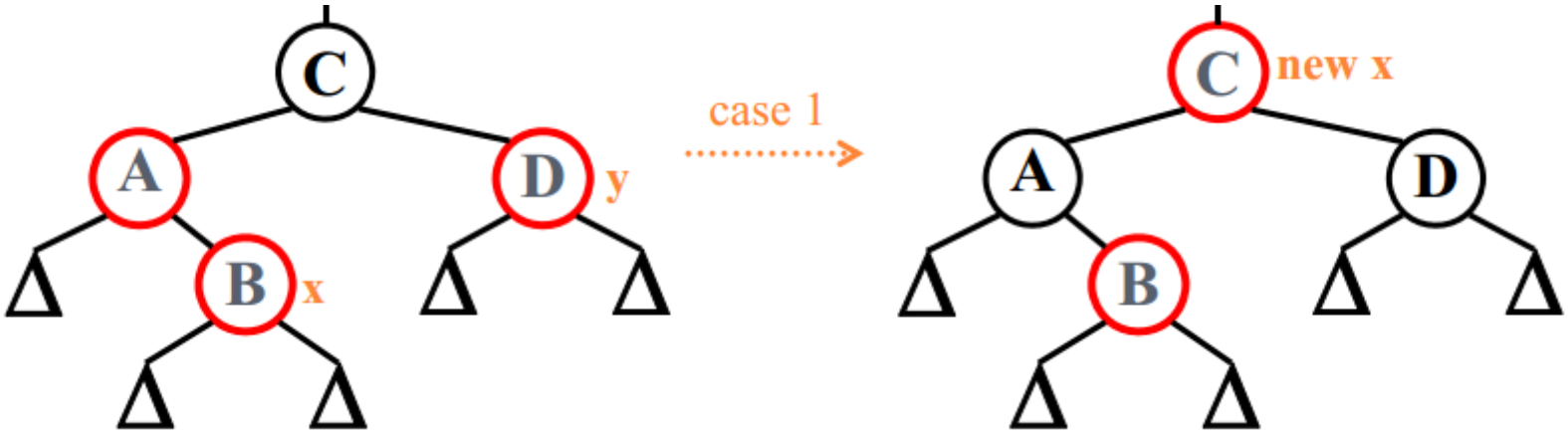


插入

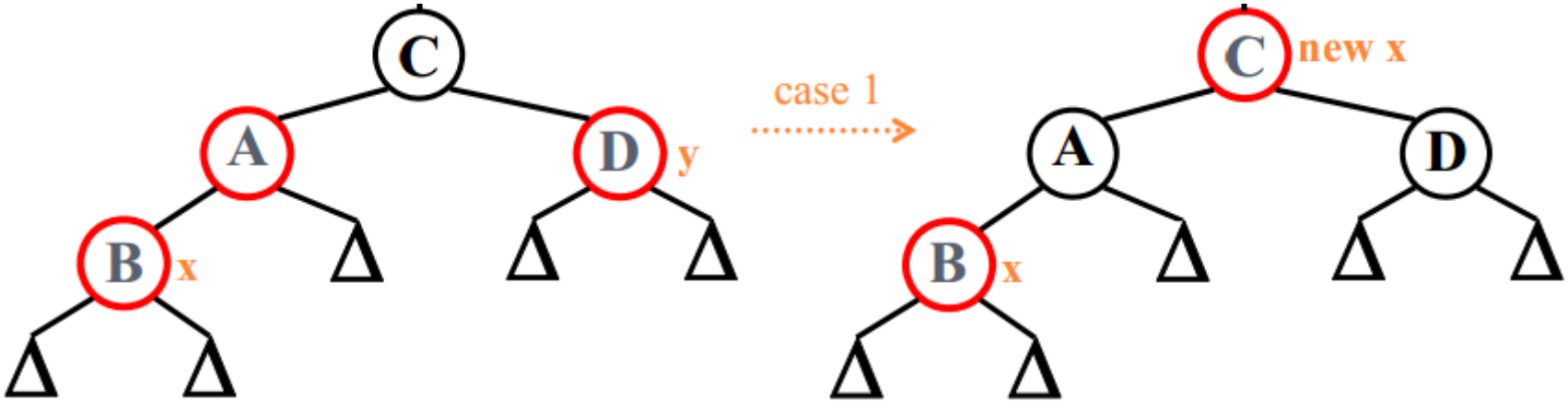
- Case 1：Z的父亲是红色，叔叔是红色
- Z的父亲是红色，叔叔是黑色
  - Case 2：z是左节点
  - Case 3：z是右节点

Case 1

- If（y->color == RED)
  - X->p->color = black
  - X->u->color = black
  - X->p->p->color = red

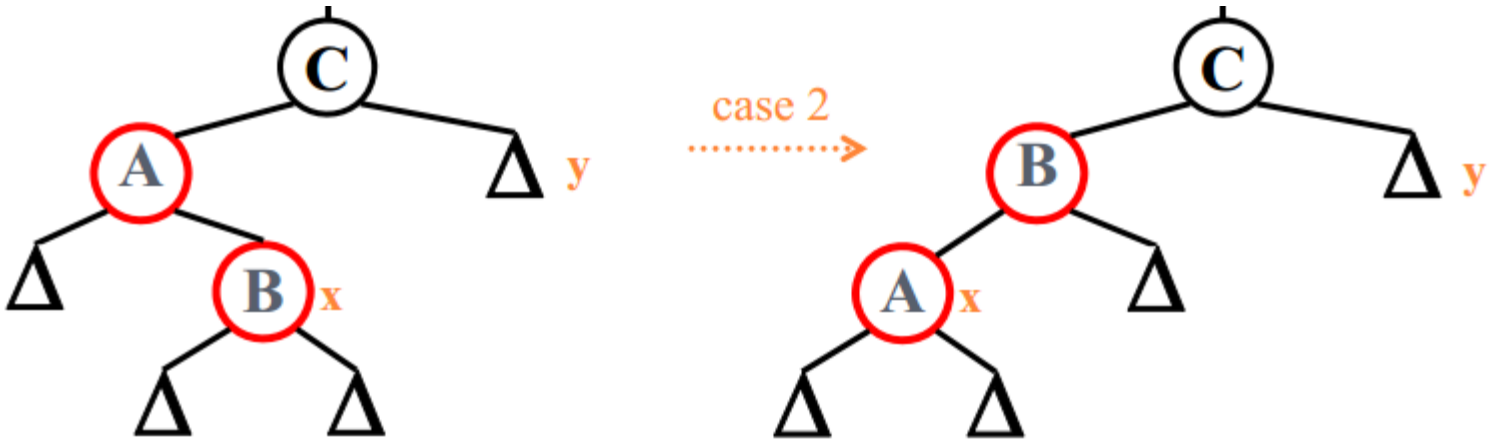


Case 1 cont.



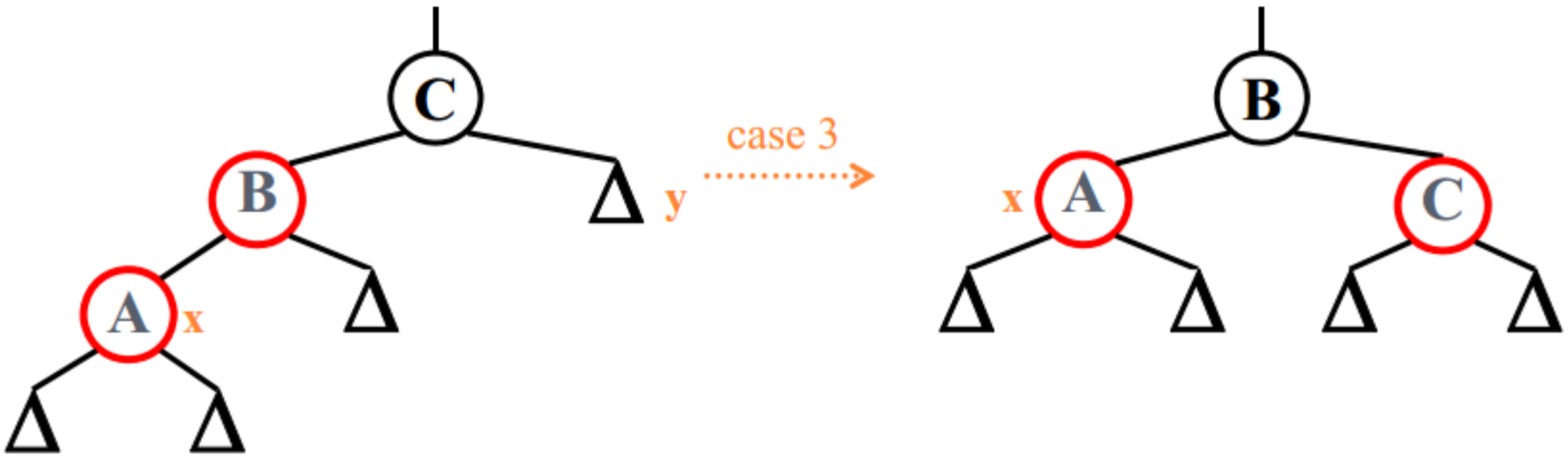
Case 2

- If(x=x->p->right)
  - x=x->p;
  - Rotate-left(x)
  - Continue case3



Case 3

- X->p->color = Black
- X->p->p->color = Red
- Right-rotate(x->p->p)



插入伪代码

```
treeInsert(x);
x->color = RED;
// Move violation of #3 up tree, maintaining #4 as invariant:
while (x!=root && x->p->color == RED)
if (x->p == x->p->p->left)
    y = x->p->p->right;
    if (y->color == RED)
        x->p->color = BLACK;
        y->color = BLACK;
        x->p->p->color = RED;
        x = x->p->p;
    else    // y->color == BLACK
        if (x == x->p->right)
            x = x->p;
            leftRotate(x);
        x->p->color = BLACK;
        x->p->p->color = RED;
        rightRotate(x->p->p);
else    // x->p == x->p->p->right
    (same as above, but with
     "right" & "left" exchanged)
```

Case 1

Case 2

Case 3

删除

- Case 1：x的兄弟w是红色
- Case 2：x的兄弟是黑色的，而且w的两个孩子都是黑色
- Case 3：x的兄弟w是黑色的，w的左孩子是红色的，右孩子是黑色
- Case 4：x的兄弟w是黑色的，w的右孩子是红色的

键树

- 手机九宫格
  - 每个数字对应三个字符
- 找一个单词来对应的数字，比如computer，对应的是26678837
- 找一组数组对应的单词，比如26678837对应的是computer

问题1

- 查找单词对应的数字？

问题2

- 查找数字对应的单词
  - 字典如何构建
  - 构建数字-单词字典

英文字典构建

- 字母有26个
- 单词的长度不等
- Hash方法
  - 用Map数据结构实现
  - 不足

英文字典构建cont.

- 键树
  - 数字查找树
  - 节点个数不同，孩子节点数大于2时，称作为Trie树
- 存储
  - 字母标记一个单词？
  - 单词很长，自单词？

数字查找单词

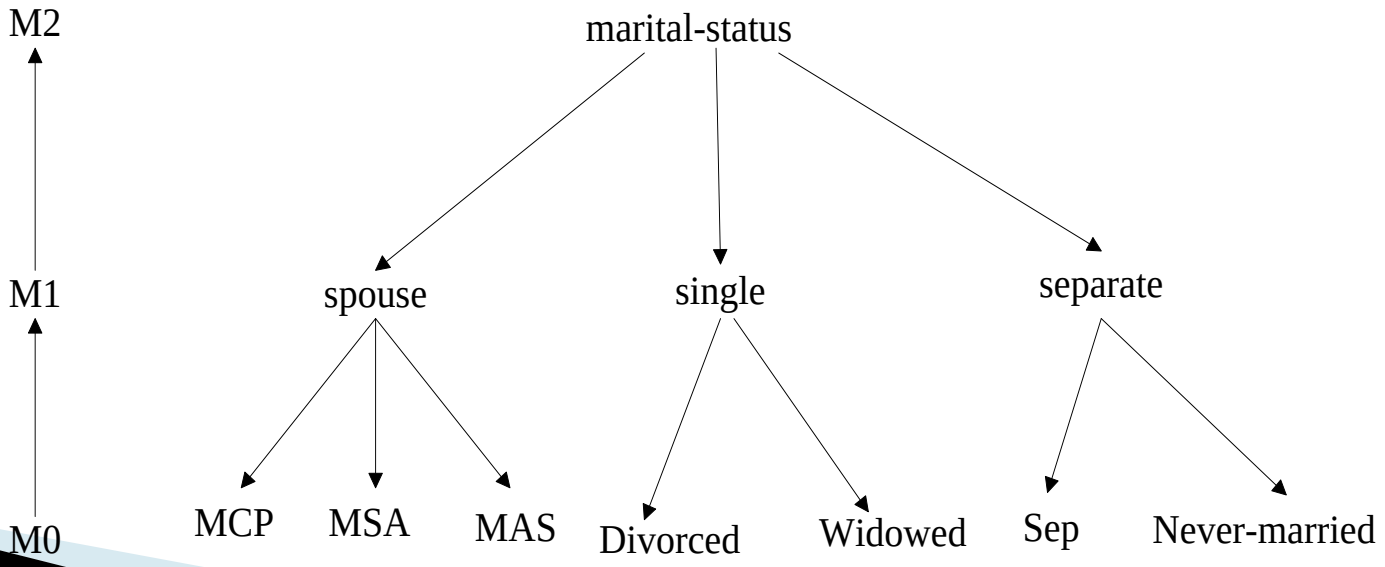
- 新建一个字典
- 新建数字-单词字典
- 剪枝

键树的应用

- 优点：
  - 最大限度地减少不必要比较的次数
  - 记录比较的过程
- 搜索引擎
  - 统计字符串出现的次数
  - 排序字符串

泛化分层（Generalization）

- 元素之间存在层次关系
  - 比如分类关系、包含关系
- 数据发布
  - 隐私保护
  - 模糊数据



K–匿名（了解）

- Latanya Sweeney等人提出著名的k-匿名的隐私保护方法，该模型要求任意记录至少有k-1个与之相同的记录，从而减少链接攻击所导致的隐私泄露
- 准标示符
- 敏感属性

高级语言中的树

- 没有指针
- 不采用二叉链表
- 孩子节点（List）

参考文献

- 数据结构-严蔚敏
- 算法导论-Thomas