

# 基础作业

---

## 1 简述二分查找的流程，说明其时间复杂度

- 二分查找的流程，假设查找的区间为  $(0, high)$ ，数组为  $R[high]$ 
  - 取中点  $mid$  的位置  $(0+high) / 2$
  - `if( $R[mid] > K$ )`，则取左半边  `$mid = (0+mid) / 2$`
  - `else if( $R[mid] < K$ )`，则取右半边  `$mid = (mid+high) / 2$`
  - `else` 如果相等，则说明找到了待查值，返回  $mid$  的值
  - 如果取到当前查找区为空，则说明不存在  $K$ ，查找失败
- 时间复杂度
  - `$O(\log n)$`

## 2 简述哈希查找的流程，说明其时间复杂度

- 哈希查找的流程
  - 确定哈希公式
  - 创建一个空数组
  - 通过将元素传入哈希公式得到返回值，凭借返回值.插入哈希数组，返回值作为数组下标  $ret \in (0 \sim \maxkey - 1)$
  - 将带插入的数据传入哈希函数，得到返回值
  - 检查数组内容是否有对应的哈希值

## 3 什么是二叉排序树？如何插入二叉排序树？

- 二叉排序树
  - 对于任意一棵子树，左子树的最大值小于根节点小于右子树的最小值。
  - 若左子树不空，则左子树上所有结点的值均小于它的根结点的值。
  - 若右子树不空，则右子树上所有结点的值均大于它的根结点的值。
- 如何插入一个二叉排序树
  - 先当作叶子节点直接插入树种
  - 然后遍历二叉树，找出离被插结点最近的父亲结点。
  - 若插入值大于父亲节点，则直接交换插入值和父亲节点，递归向上插入
  - 若此时树为空。则首先单独生成根结点。

## 4 对于1, 2, 3三个元素，二叉排序树有多少种不同的形式？

- 对于1, 2, 3三个元素，二叉排序树有5种形式，分别为



## 5 如何删除一个二叉排序树的元素？

- 如果是叶子节点，直接删除
- 如果有孩子节点
- (需要找到最接近节点值的叶子节点替换之后删除叶子)
  - 找到孩子的左子树最左边或者右子树最右边（最接近的值）
  - 用叶子节点覆盖源节点
  - 删除叶子
- 仅有左或者右子树的结点：删除结点后，将它的左子树或者右子树整个移动到删除结点的位置。

## 6 二叉排序树中，什么是左旋，什么是右旋？

- 左旋(右孩子为轴,当前结点左旋) 根 → 左孩子
- 右旋(左孩子为轴,当前结点右旋) 根 → 右孩子

## 7 红黑树的特征有哪几个？

- 节点必须是红 || 黑（节点必须有颜色）
- 根节点必须是黑色
- 每个红色节点必须有两个黑色的孩子
- 两个红色节点不能连续
- 根到叶子经历的黑色结点数要相同

## 8 整理红黑树插入的几种情况

插入节点一开始为红叶子节点。

- 若父红叔红 → 父叔染黑爷染红（相当于在爷的位置插入新节点，之后递归判断）
- 如果根是红 → 直接染黑
- 如果父是黑 → 不变（直接插入）

\*9 整理红黑树删除的几种情况

- 删除的节点是红色的
  - 若无孩子节点，则直接删除
  - 若有，则交换右孩子（若无则交换左孩子）后直接删除
- 删除的节点是黑色的
  - 如果删除的节点的右孩子不为空，则直接节点和右交换后删除
  - 若删除的节点的右孩子不空，则直接节点和右交换后删除需要判断叔节点的颜色
    - 若为红，则将树左旋，将叔节点变为根节点并染黑，继续向上判断

