# 数据结构

* [1. 各种排序算法的性质](#1-各种排序算法的性质)
* [2. 二叉搜索树、AVL树和红黑树的复杂度](#2-二叉搜索树avl树和红黑树的复杂度)
* [3. 红黑树的性质](#3-红黑树的性质)
* [4. 图的基本术语](#4-图的基本术语)
* [5. B树的性质](#5-b树的性质)
* [6. B+树的性质](#6-b树的性质)
* [7. B树和B+树的区别](#7-b树和b树的区别)

## 1. 各种排序算法的性质

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 算法种类 | 最好 | 平均 | 最坏 | 空间复杂度 | 是否稳定 |
| 直接插入排序 | O(n) | O(n^2) | O(n^2) | O(1) | 是 |
| 冒泡排序 | O(n) | O(n^2) | O(n^2) | O(1) | 是 |
| 简单选择排序 | O(n^2) | O(n^2) | O(n^2) | O(1) | 否 |
| 快速排序 | O(nlogn) | O(nlogn) | O(n^2) | O(logn) | 否 |
| 堆排序 | O(nlogn) | O(nlogn) | O(nlogn) | O(1) | 否 |
| 二路归并排序 | O(nlogn) | O(nlogn) | O(nlogn) | O(n) | 是 |
| 基数排序 | O(d(n+r)) | O(d(n+r)) | O(d(n+r)) | O(r) | 是 |

## 2. 二叉搜索树、AVL树和红黑树的复杂度

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 树种类 | 空间 | 空间 | 查找 | 查找 | 插入 | 插入 | 删除 | 删除 |
|  | 平均 | 最坏 | 平均 | 最坏 | 平均 | 最坏 | 平均 | 最坏 |
| 二叉搜索树 | O(n) | O(n) | O(logn) | O(n) | O(logn) | O(n) | O(logn) | O(n) |
| AVL树 | O(n) | O(n) | O(logn) | O(logn) | O(logn) | O(logn) | O(logn) | O(logn) |
| 红黑树 | O(n) | O(n) | O(logn) | O(logn) | O(logn) | O(logn) | O(logn) | O(logn) |

## 3. 红黑树的性质

红黑树在满足二叉搜索树的要求外，还需满足如下要求：

* 节点是红色或黑色。
* 根是黑色。
* 所有叶子都是黑色（叶子是NIL节点）。
* 每个红色节点必须有两个黑色的子节点。（从每个叶子到根的所有路径上不能有两个连续的红色节点。）
* 从任一节点到其每个叶子的所有简单路径都包含相同数目的黑色节点。

## 4. 图的基本术语

* 简单图：一个图G如果不存在重复边，也不存在顶点到自身的边，则称图G为简单图。
* 无向完全图：在无向图中，如果任意两个顶点之间都存在边，则称该图为无向完全图。含有n个顶点的无向完全图有n(n-1)/2条边。
* 有向完全图：在有向图中，如果任意两个顶点之间都存在方向互为相反的两条弧，则称该图为有向完全图。含有n个顶点的有向完全图有n(n-1)条有向边。
* 连通：在无向图中，若从顶点v到顶点w有路径存在，则称v和w是连通的。
* 连通图：若图G中任意两个顶点都是连通的，则称图G为连通图。
* 连通分量：无向图的极大连通字图称为连通分量。
* 强连通图：在有向图中，若从顶点v到顶点w和从顶点w到顶点v之间都有路径，则称这两个顶点是强连通的。若图中任何一对顶点都是强连通的，则称此图为强连通图。
* 强连通分量：有向图中的极大强连通字图称为有向图的强连通分量。
* 生成树：连通图的生成树是包含图中全部顶点的一个极小连通字图。

## 5. B树的性质

B树是一种多路平衡查找树，B树中所有结点的孩子结点数的最大值称为B树的阶，通常用m表示。一颗m阶B树，或为空树，或为满足下列特性的m叉树：

* 树中每个结点至多有m颗子树（即至多含有m-1个关键字）。
* 若根结点不是叶子结点，则至少有两颗子树。
* 除根结点外的所有非叶结点至少有⌈m/2⌉颗子树（即至少含有⌈m/2⌉-1个关键字）。
* 所有的非终端结点中有以下数据：[n, P0, K1, P1, K2, P2, ..., Kn, Pn]。其中，Ki（i = 1, 2, ..., n）为结点的关键字，且有K1 < K2 < ... < Kn；Pi（i = 0, 1, ..., n）为只想子树根结点的指针，且指针Pi-1所指子树中所有结点的关键字均小于Ki，Pn所指子树中所有结点的关键字均大于Kn，n（⌈m/2⌉ - 1 <= n <= m -1）为结点中关键字的个数。
* 所有的叶结点都出现在同一层次上，并且不带信息（可以看做是外部结点，实际上这些结点不存在，指向这些结点的指针为空）。

## 6. B+树的性质

B+树是B树的一种变形树，一颗m阶B+树应满足下列条件：

* 每个分支结点最多有m颗子树（子结点）。
* 根结点或者没有子树，或者至少有两颗子树，其他每个分支结点至少有⌈m/2⌉颗子树。
* 根结点的子树个数与关键字个数相等。
* 所有叶结点包含全部关键字及其只想相应记录的指针，而且叶结点中将关键字按大小顺序排列，并且相邻叶结点顺序链接起来，所有叶结点在同一层。
* 所有分支结点（可以看成是索引的索引）中仅包含其子树（即下一级的索引块）中关键字的最大值及指向其子树的指针。

## 7. B树和B+树的区别

* 在B+树中，具有n个关键字的结点只含有n颗子树，即每个关键字对应一颗子树；而在B树中，具有n个关键字的结点含有n + 1颗子树。
* 在B+树中，每个非根结点关键字个数n的范围是⌈m/2⌉ - 1 <= n <= m - 1（根结点：1 <= n <= m - 1）。
* 在B+树中，所有非叶结点仅起到索引作用，即结点中的每个索引项只含有对应子树的最大关键字和指向该子树的指针，不含有该关键字对应记录的存储地址。
* 在B+树中，叶结点包含了全部关键字，即其他非叶结点中的关键字包含在叶结点中；而在B树中，叶结点包含的关键字和其他结点包含的关键字是不重复的。