

常用 search 算法

常用的 search（给对象找位置）方法有：顺序查找、二分查找、二叉搜索树、Hash 表

顺序查找

数据本身没有特点，乱序，需要遍历、对比来获取对象位置。

时间复杂度为： $O(N)$ （从头到尾一个一个找，简单粗暴，但时间复杂度不好）

二分查找

数据必须是有序的，先找到中间值，将被查对象与数据集中间值比较，大于则在后半段继续，小于则在前半段继续。如大于，在后半段继续取其中间值，重复以上操作，直至找到对象。

时间复杂度： $O(\log N)$ 。

（二分查找基础且重要，是一些较复杂算法的基础，算法思想很常用，面试时也是常考的）

二叉搜索树

二叉搜索树是二分查找的二叉树实现，二叉搜索树每个节点都有作为搜索依据的关键码，所有节点的关键码互不相同。左子树（若存在）上的所有节点的关键码都小于根节点的关键码，右子树（若存在）上的所有节点的关键码都大于根节点的关键码，左子树和右子树都是二叉搜索树。

时间复杂度：二叉搜索树— $O(h)$ 、平衡二叉树— $O(\log N)$

（ h 为树高度，二叉树越平衡，时间复杂度越好，要是树太偏，时间复杂度就会很不好）

Hash 表（哈希表|散列表）

要做的就两件事，1、散列函数；2、解决冲突；

散列函数（直接定址）的构建有许多方法，最常用的是取余数法。

冲突：一个位置已有元素，另一个元素经过散列计算得出也要放在这个位置，即发生了冲突。

解决冲突的方法有：开放地址法和分离链接法。

开放地址法就是遇到冲突就换位置，按换位置的方法又分为线性法、平方法等；

分离链接法是在冲突位置构造链表，所有冲突的元素都用链表连起来，查找时先找到冲突位置，再在链表里找。

时间复杂度： $O(\log N)$

（总的来说，“直接定址”与“解决冲突”是哈希表的两大特点。）

注：1、静态查找：顺序、二分查找，动态查找：树查找、Hash 表；

2、动态查找当元素比较复杂时用 Hash 表，比如变量管理（定义—插入、调用—查找）、（QQ）账号登录等。即将比较的叫复杂的元素（如字符串）转换为数字，就好处理了；

3、顺序查找——>分块查找；二分查找——>插值查找、斐波那契查找；树表查找：二叉搜索树——>平衡二叉树：2-3 搜索树、红黑树、B 树|B+树；（这些查找难度较大，之前听讲座有个大神也说“面试官要是现场让你写红黑树，那就是搞事情……”，不过，人家就是搞事情呢？所以，练吧！）

（七大查找算法，大神更为详细的讲解，戳~~~）

http://www.cnblogs.com/maybe2030/p/4715035.html#_label0