数据结构

叶茂林 2024年4月8日星期一

**红黑树**

自平衡的二叉搜索树，查找O(logn)，根节点和叶子节点为黑，红节点的子节点为黑，简单路径上的黑节点数目相同

**哈希冲突**

线性探测、二次探测、开链接、再哈希

**Vector扩容**

性能问题，需要内存分配和元素复制，原迭代器失效

**map的insert和下标访问区别**

下标访问会查找元素的迭代器，如果没有找到，会使用默认构造函数构造一个，然后返回value引用

Insert会调底层红黑树的方法查找要插入元素的位置，如果可以找到那么调用红黑树的插入操作，如果没有找到说明已经存在key，插入失败

这两个的使用区别在于对于相同的key，insert不会更新map，而下标会覆盖原有的值

**Vector赋值实现**

首先确定赋值的对象是否和当前对象不同，再确定是否使用相同的分配器，新的容量是否超过当前容量，如果超过申请一块更大的，然后释放原本空间，复制新元素，更新容量和大小

**Vector删除元素**

如果是erase的话，把要删除位置后面的元素往前移动，如果是pop\_back的话，只需要将容器size减一