07-A1

7.二叉搜索树

概述

循关键码访问

There's nothing in your head the sorting hat can't see. So try me on and I will tell you where you ought to be.

邓俊辉

deng@tsinghua.edu.cn

查找

- 按照事先约定的规则，从数据集合中找出符合特定条件的对象
- 对于算法的构建而言，属于最为基本而重要的静态操作
- 很遗憾，基本的数据结构并不能高效地兼顾静态查找与动态修改

基本结构	查找	插入/删除
无序向量	O(n)	O(n)
有序向量	O(logn)	O(n)
无序列表	O(n)	O(1)
有序列表	O(n)	O(n)

- 那么，能否综合现有方法的优点？如何做到？

循关键码访问

❖ 数据项之间，依照各自的关键码彼此区分

call-by-key

❖ 为此，关键码之间必须支持

- 大小比较与
- 相等比对

❖ 数据集合中的数据项

统一地表示和实现为词条entry形式

词条

```
❖ template <typename K, typename V> struct Entry { //词条模板类
    K key; V value; //关键码、数值
    Entry( K k = K(), V v = V() ) : key(k), value(v) {}; //默认构造函数
    Entry( Entry<K, V> const & e ) : key(e.key), value(e.value) {}; //克隆
  // 比较器、判等器（从此，不必严格区分词条及其对应的关键码）
    bool operator< ( Entry<K, V> const & e ) { return key < e.key; } //小于
    bool operator> ( Entry<K, V> const & e ) { return key > e.key; } //大于
    bool operator==( Entry<K, V> const & e ) { return key == e.key; } //等于
    bool operator!=( Entry<K, V> const & e ) { return key != e.key; } //不等
  };
```