Lab4 - BBST Report

2021201709 李俊霖

1. 问题1

1.1.AVL树

基本思想:

- 一种二叉搜索树，树的左右子树都是AVL树，树的左右子树高度之差（平衡因子）的绝对值不超过1。
- 通过旋转调整来保持平衡因子符合规则。

基本功能实现思路:

- 插入:
  - 按照二叉搜索树的方式插入新节点，大的往右，小的往左。插入之后需要检查是否满足平衡条件，进行重平衡操作。
  - 重平衡过程中有单旋双旋等四种旋转操作，此处不再一一赘述。
- 删除:
  - 先按照二叉搜索树的基本删除操作进行删除，分为右孩子为空，左孩子为空，左右均不为空三种情况。
  - 删除之后需要进行重平衡操作。
- 查找:
  - 若为空树，查找失败，返回nullptr;
  - 若x小于当前结点的key，在该结点的左子树中继续查找;
  - 若x大于当前结点的key，在该结点的右子树中继续查找;
  - 若x等于当前结点的key，查找成功。
- 查询x的排名：对该值的数字进行查找，大的往右，小的往左，最终命中是计算结果为左子树的size+1。
- 查询排名为x的数：取中序遍历序列的第x个值。
- 求x的前驱:
  - 找到x值对应的节点的前驱节点，分为两种情况
    - 左子树不为空，则在左子树的最右下节点。
    - 左子树为空，则向右上走到祖先节点，再往上的一个父节点为前驱节点。
  - 输入数据有重复，所以假如找到的前驱节点值与key相等，再以此节点继续寻找前驱节点，直到找到比key值小的或者null为止。
- 求x的后继：与求前驱节点思路正好相反，不在一一赘述。

1.2.红黑树

基本思想：

- 一种二叉搜索树，每个节点增加颜色标记（红或黑）。
- 红黑树保证最长路径不超过最短路径的二倍，因而近似平衡
- 最短路径就是全黑节点，最长路径就是一个红节点一个黑节点，当从根节点到叶子节点的路径上黑色节点相同时，最长路径刚好是最短路径的两倍
- 基本特性：
  - 根为黑
  - 叶子节点（外部节点，空节点）是黑色，这里的叶子节点指的是最底层的空节点（外部节点）。
  - 从根节点到叶子节点的所有路径上不能有 2 个连续的红色节点。
  - 从任一节点到叶子节点的所有路径都包含相同数目的黑色节点。

基本功能实现思路：

- 插入：
  - 基本步骤：
    - 按照二叉搜索的树规则插入节点
    - 检查红黑树的性质是否完好（新节点的默认颜色是红色）
    - 检查与调整方法：
      - 如果父亲节点为黑色，则无需调整。
      - 如果父亲节点为黑色，双红，要调整，看插入结点的叔叔来分类讨论。
- 删除：
  - 删除红结点，无影响。
  - 删除1度黑结点，把其孤立的儿子树挂到父结点上，并把孤儿子树根结点染黑（度为1的黑色节点，唯一子孩子，一定是红色）。
  - 删除2度黑结点，转换成删除度为1或度为0的情况。
  - 删除0度黑结点产生双重黑NIL结点。删除调整去除双重黑。
- 查询x的排名：对该值的数字进行查找，大的往右，小的往左，最终命中是计算结果为左子树的size+1。
- 查询排名为x的数：根节点开始递归，找到地k大对应的位置。
- 求x的前驱：
  - 找到x值对应的节点的前驱节点，分为两种情况
    - 左子树不为空，则在左子树的最右下节点。
    - 左子树为空，则向右上走到祖先节点，再往上的一个父节点为前驱节点。
  - 输入数据有重复，所以假如找到的前驱节点值与key相等，再以此节点继续寻找前驱节点，直到找到比key值小的或者null为止。
- 求x的后继：与求前驱节点思路正好相反，不在一一赘述。

2. 问题2

基本思路：

- 本问题需要进行区间翻转，涉及到交换，想到splay树中的左右子树旋转和交换。
- 对 1 - n 建一个平衡树，该序列即为平衡树的中序遍历，最终需要输出的也是这棵树的中序遍历。
- splay树反转区间的思路为：如果需要翻转的区间为l到r，先将l-1转到根，再将r+1转到l-1的右孩子。这个时候可以发现，l到r这个区间内部在树中即为r+1的左子树。对此时只要交换所有节点的左右子树，即可实现对l到r区间的翻转。
- 用tag标记r+1的左孩子，把需要翻转的子树的根标记，之后访问到该点时交换其左右子树和将标记下传。

3.总结与收获

- 通过这次lab，笔者对各种平衡树的基本思路有了更深刻的理解和认识。比如对于问题二，笔者刚开始时没有马上想到用平衡树的思路去解决问题，经过学习和思考后发现，这种交换与平衡树的交换左右子树有异曲同工之妙，遂深感平衡树的用处之大！
- 由于树的代码写得不多，笔者在调试的过程中花费了大量的经精力，尤其是红黑树和AVL树部分，在不断的调试和试错中，对树的基本操作、遍历、前驱后继等也更熟悉。

4.参考资料

1. Splay处理区间操作——翻转操作(Reverse)
2. 特别浅地浅谈Splay
3. C++红黑树
4. AVL树的详细实现(C++)
5. 《数据结构(c++语言版)》邓俊辉编著-清华大学出版社