实现二叉排序树建立和检索

（采用Visual Studio 2017 15.7.2 Windows SDK 10.0.17134.0）

这可能是最后一次上机作业的一部分了，不过因为之前VS装的插件比较多，还装在移动硬盘上，最近出现了文本编辑器无法显示文本的bug，导致重新安装了VS，sdk版本号有所变化。

总的来说这次作业想很快完成其实直接一个个简单插入就可以完成任务，但是毕竟讲了AVL树，就觉得还是试一试比较好，结果就主要把精力都放在对AVL树的处理上了。在码代码时发现这个内容有很多细节需要注意，不然就会出各种错误，导致debug了很久。。。

0、程序要完成的内容

实现二叉排序树建立和检索，自己加上要求，要求是AVL树。（最佳二叉树就算了吧。。。有点难实现了，权重也不好随便设置）

1、最终程序的结构是这样的：

首先在整个二叉排序功能类的构造函数中提示用户输入要排序的内容，然后按输入构造出二叉排序树，依照用户的需求进行查找，重建以及退出。重建通过将循环断开从而将操作类移出作用域而自动调用析构函数，然后用外层循环再次重新构造。退出则直接把两个循环都跳出（通过控制两个循环的布尔型变量完成）。

因为想练习使用顺序存储结构，也因为上次Haffman编码的影响，使用的是动态分配内存的大小刚好够的顺序表结构，因此并不能很好地处理临时添加和减少条目。

（这也是设计上的一个问题，因为用的是c++ 的new[] 和delete[] 来实现，所以不能用纯C里面的realloc函数，而拷贝构造函数更是会大大降低效率，故没有此功能，但在二叉树构造时实际上也是一个个条目插入到树里面的，所以其实原始的功能还是有的，只是没有单独作为功能可供用户调用罢了）

在用户输入中，程序是先把内容一次性读入到内存中，然后再进行排序的。但如果使用的是链接存储结构，可能一边输入一边构造比较合理，如果先输入再构造可能需要一个单链表结构，存储结构的变化会比较奇特，也不合适。

2、一些遇到的问题和解决方法

在写程序的过程中，因为顺序表里面表示排序码的变量是int型，表示下标的变量是long long 或者size\_t型（long long是为了返回-1，其实throw也可以做到，但是那个不是很好，也不是应该做的，而且还要自己写一个特殊的错误类），所以很容易把代码写串，尤其是在一层层访问时（相当于链接结构的多层‘->’运算符时），一大长串的data[data[data[…..].rchild].lchild].parent实在是很精神污染，并且rchild 和lchild很容易写反。

在这么多东西都需要精力时，最小不平衡子树的调整的掌握不是很熟练就体现出来了。个人感觉不看着算法的示意图就想不清楚了，而且各个下标存储的逐个改变的顺序还要搞正确，否则就会出现不是二叉树的结构，或者子节点的父节点不是这个节点一类的问题

这些问题都是慢慢调试，检查才处理掉了的。尤其是当时写的时候在凌晨，就很不在状态，花了很久。。。（当时还有过某些处理的路径上直接用子节点的深度当作父节点深度的情况（程序中使用了深度直接判定平衡，而没有用平衡因子，实际上没有区别））

具体这些错误应该git的版本管理都有体现，但是之前我把各个数算课的代码都整合到了一个repo里面，所以可能并不方便直接附带.git文件夹的了，但是在提交作业之后将会把最新的代码同步到github上我的版面上，应该也算可以吧。。。

地址：<https://github.com/shangjiaxuan/Class-Projects>

3、一些可以改进的地方

首先就是关于动态插入的问题了，实际上的确，使用链接存储是一个更好的方法，毕竟二叉排序树在讲课时就说过是“动态字典”的合理表示方式。二叉排序树的优点就在于可以方便插入，删除，并且合适的树可以减少检索需要的运算复杂度。使用顺序表挑战自我，有点舍本逐末了，并且并没有很好地提高任何效率。

然后就是对于先录入数据再计算的方法了。这种做法在文件不稳定（硬盘突然摘下来一类）时可能有用，但实际上提前全部载入内存并没有什么意义，而且这样的方法并不能很好地进行改造成其他存储结构（链接存储），因此可能在构造函数里同时进行录入和排序。但这样代码量会大大增加（因为有两种可能的输入方式，手动和文件），而且一个函数的长度会大大增加，大大提高代码的维护难度。可能需要更好的解决方案（更合适地分割成小一些的功能性的函数。）