数据结构实验报告二

实验名称 树及其应用

### 实验目的及实验要求

1. 通过实验掌握二叉树的两种基本的存储结构，及二叉树的建立、遍历，并加以应用。
2. Huffman树的建立和编码。

### 2实验内容及实验步骤（附运行结果截屏）

1. 按先序次序输入二叉树中结点的值，建立一棵以二叉链表作存储结构的二叉树，然后按先序、中序、后序顺序分别遍历这棵二叉树，并完成二叉树的相应信息的统计（如各种结点数目、二叉树的高度等）；
2. 建立一棵二叉排序树，并对其进行先序、中序、后序遍历。
3. 应用队列结构实现二叉树的层次遍历。
4. 设计一个完整的编码系统：针对一篇文档，统计各个字符的出现次数，并为其设计Huffman编码。

注：（1）～（2）必做，（3）～（4）选做。

实验步骤：

先仍然是先编写类模板，实现各自的基础结构，之后按照要求编写适当的函数方法（公共接口），最后完成封装。编写主函数直接调用即可。

核心代码：

//BirTree.h 二叉树

1. class BinSTree;
2. class BinStreeNode
3. {
4. public:
5. KeyType key;
6. BinStreeNode \*lchild;
7. BinStreeNode \*rchild;
8. BinStreeNode()
9. {
10. lchild=NULL;
11. rchild=NULL;
12. }
13. };
14. class BinSTree
15. {
16. public:
17. BinStreeNode \*root;
18. BinSTree()
19. {
20. root=NULL;
21. }
22. ~BinSTree()
23. {
24. //delete root;
25. }
26. BinStreeNode \*BSTreeSearch(/\*BinStreeNode \*bt,\*/KeyType k,BinStreeNode \*&p);
27. void BSTreeInsert(KeyType k);
28. int BSTreeDelete(KeyType k);
29. void BSTreePreOrder(BinStreeNode \*bt);
30. void BSTreeInOrder(BinStreeNode \*bt);
31. void BSTreeAfOrder(BinStreeNode \*bt);
32. void BSTreeLevelTraverse(BinStreeNode \*bt);//层次遍历
33. bool IsEmpty()
34. {
35. return root==NULL;
36. }
37. };

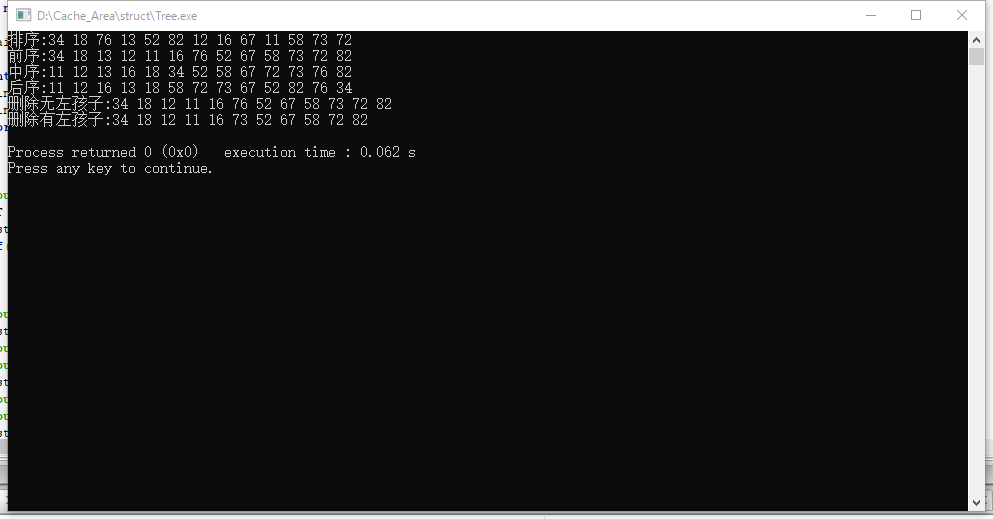
//中序遍历

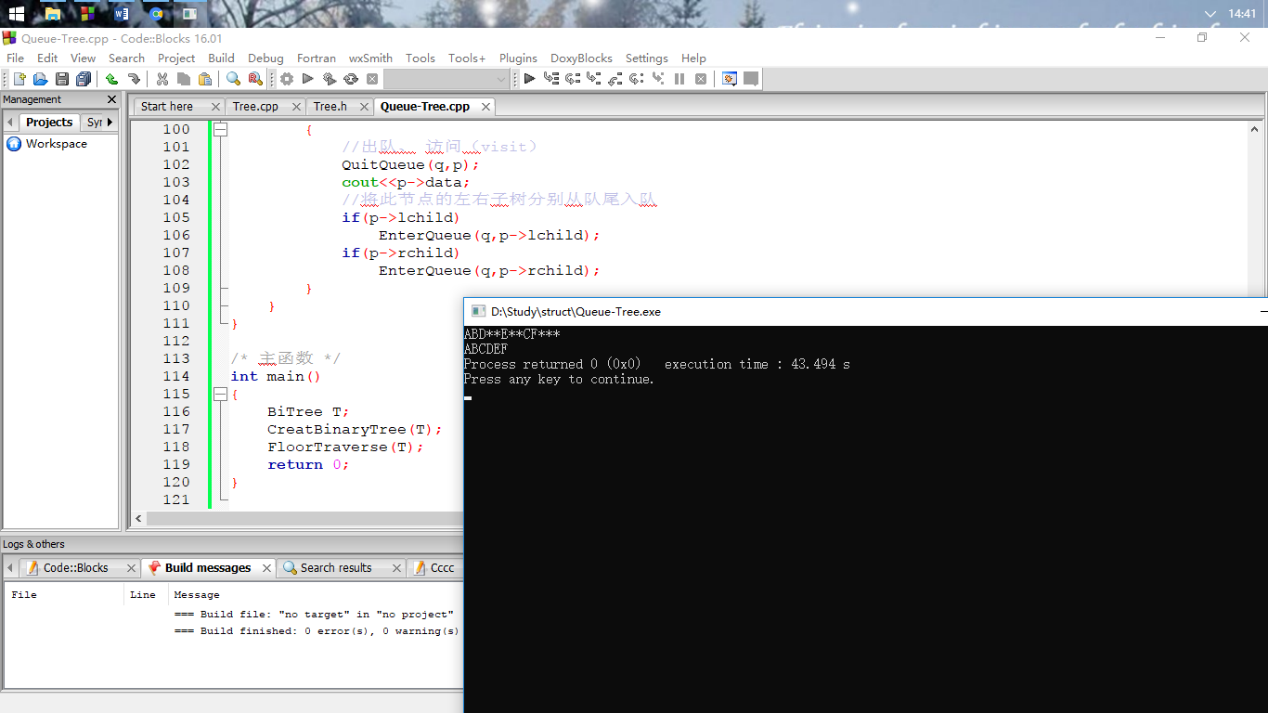
1. void BinSTree::BSTreeInOrder(BinStreeNode \*bt)
2. {
3. if(bt)
4. {
5. BSTreeInOrder(bt->lchild);
6. cout<<bt->key<<" ";
7. BSTreeInOrder(bt->rchild);
8. }
9. }

//二叉排序树的删除

1. int BinSTree::BSTreeDelete(KeyType k)
2. {
3. BinStreeNode \*f,\*p,\*q,\*s;
4. p=root;
5. f=NULL;
6. while(p&&p->key!=k)
7. {
8. f=p;
9. if(p->key>k)
10. p=p->lchild;
11. else
12. p=p->rchild;
13. }
14. if(!p)
15. return 0;
16. if(p->lchild==NULL)
17. {
18. if(f==NULL)
19. root=p->rchild;
20. else if(f->lchild==p)
21. f->lchild=p->rchild;
22. else
23. f->rchild=p->rchild;
24. delete p;
25. }
26. else
27. {
28. q=p;
29. s=p->lchild;
30. while(s->rchild)
31. {
32. q=s;
33. s=s->rchild;
34. }
35. if( q == p )
36. q->lchild = s->lchild;
37. else
38. q->rchild = s->lchild;
39. p->key = s->key;
40. delete s;
41. }
42. return 1;
43. }

截图：





3 实验体会（实验遇到的问题及解决方法）

这一次要比第一次试验的进度快得多，因为大部分的操作和方式可以借鉴第一次试验，。另外栈和队列本身操作和结构也比较简单。

遇到的问题是，在判断括号成对的时候，对多重多种括号的处理方式，这一点是借鉴了网上一篇博客的思想。最后，之前的代码精简和变量命名有了自己系统的方式。