通讯模拟问题( Haffman编码和译码)

（采用Visual Studio 2017 15.6.4 Windows SDK 10.0.16299.0）

这是这学期第五次上机作业了。整体来说只要知道了算法的原理，实现起来还是不难的。在有一个地方貌似还可以优化一下，但是因为水平原因，不是会做，所以没有在程序中体现出来。

0、程序要完成的内容

实现Haffman树的建立和运用Haffman树的编码和译码（指定输入输出文件，目前没有整合到之前写的shell上面，所以暂时不能切换工作目录（其实切换目录也不难，而且c++ 17标准库好像是有这个函数的））。

1、最终程序的结构是这样的：

首先构建一个Haffman类，在构造函数中提示用户为其输入一个生成树的样本。读取样本，统计各个字符出现的次数，记录在树的顺序结构的前256个空间，下标对应这个字符（要把char转换成unsigned char，因为VC编译器默认char是signed char（据说和arm linux不一样），这样不管怎么样代码都是“portable”的）。接着把后面的255个空间作为树的不是叶的节点构造树。算法就是标准的Haffman算法。

然后进入UI，提示用户输入命令（为了方便，没有用标准的命令行格式，而是用了一行一个“词”的格式）。可以通过命令调用编码，译码操作和重建新的树（新的样品）（当然也有退出程序的功能）。

2、一些做法，想法来源和解决方法

首先，字符对应前256个节点主要目的是消除叶节点和中间节点的区别导致的需要的存储结构的不同，叶节点除了要保存加权数以外还要保存这个叶对应的字符的信息。而如果不这样处理，就只有两种解决方式，一种是用不同结构存储叶和中间节点，另一种是中间节点也保持叶的结构，但里面的字符变量不使用。这两种方法按常理分析都是不如直接像我这样的。想法的来源就是之前写词索引表时使用的“256叉树”。

树的总结点数不管权值怎么分布，总是256+（256-1），所以就没有开再多的空间了，前256个对应叶，后255个对应中间节点（其实按算法，总是按顺序一个个把中间节点加入，所以根其实总是在最后的（下标510），但仍然写了一个函数判定是否有且只有一个根，并初始化根，保证程序一定不会出错）。

关于树的建立，其实还是有debug蛮久的。主要是在查找最小权重的子树时，一开始考虑的是两个树，最小和次小一起找到，这样少找一趟。但是这样两个数的返回值的初始化会是一个问题，并且会有不稳定。最后还是就找一个了（注意这个函数要传入一个数，保证找到的不是现在正在找子节点的节点）。对于“子树”的判定，则是按没有父节点来确定的（初始父节点都是-1）。由于我们强制要求前256个节点才是叶，而一开始所有256-510下标的节点都是父节点为-1，权重为0，所以找的时候要求找到的数不能超出新添加的节点的位置（相当于手动让后面的空间成为“未分配”的状态）。

因为高电位一般是不为0，耗能的，所以规定左为0的同时就也要求了左面权值大一些。

3、总结

总的来说，我觉得还是很好地完成了要完成的任务的。以后可以考虑再进行优化，加入到我的shell里面，或者把输出的0-1码用更好的方式表示和输出（表示：不是用一个字符表示一个，而是按位运算，输出：把0-1和实际的高低电位联系起来）。