# Huffman编码（第13周布置的）

代码行数：268行

算法思想：读取文档统计每个单词出现的次数，根据次数大小进行从小到大排序去进行哈夫曼树的构建。根据哈夫曼编码去寻找对应字母的编码来转译字符。同样用这个表去翻译前四段存入文件。每次读取一个编码转译为英文单词后连接为一段后存入文档。

主要/核心函数分析：//哈夫曼树void HufCode()从第53个结点开始，寻找两个没有父亲的最小结点作为它的父亲，并更新它的值为两个孩子的和并放入数组，标记两个子节点的父亲为它。继续搜索第54个结点一直到2\*52-1。从叶节点到根节点依次往上遍历获取01序列，最后翻转得到该叶子结点的哈夫曼编码。

测试数据： 老师给的text

运行结果：

统计结果

899 135 416 356 1295 233 217 369 856 22 42 454 271 750 829 275 19 703 696 1080 283 132 123 31 138 8 22 4 21 25 7 13 15 6 24 0 0 1 22 12 14 12 0 5 12 32 8 2 11 0 0 0

转译Data Structure

000101101 0000 110 0000 0001011001 110 0110 10101 00011 110 10101 0110 100

第1段

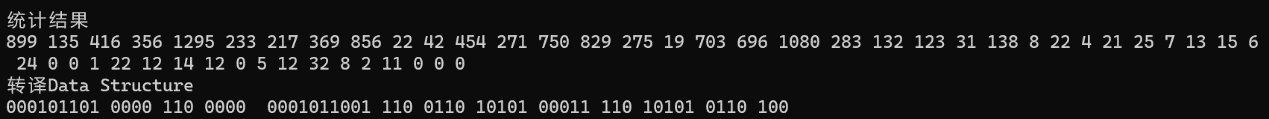
OneofthebasicrulesconcerningprogrammingisthatnoroutineshouldeverexceedapageThisisaccomplishedbybreakingtheprogramdownintomodulesEachmoduleisalogicalunitanddoesaspecificjobItssizeiskeptsmallbycallingothermodulesModularityhasseveraladvantagesFirstitismucheasiertodebugsmallroutinesthanlargeroutinesSeconditiseasierforseveralpeopletoworkonamodularprogramsimultaneouslyThirdawellwrittenmodularprogramplacescertaindependenciesinonlyoneroutinemakingchangeseasierForinstanceifoutputneedstobewritteninacertainformatitiscertainlyimportanttohaveoneroutinetodothisIfprintingstatementsarescatteredthroughouttheprogramitwilltakeconsiderablylongertomakemodificationsTheideathatglobalvariablesandsideeffectsarebadisdirectlyattributabletotheideathatmodularityisgood

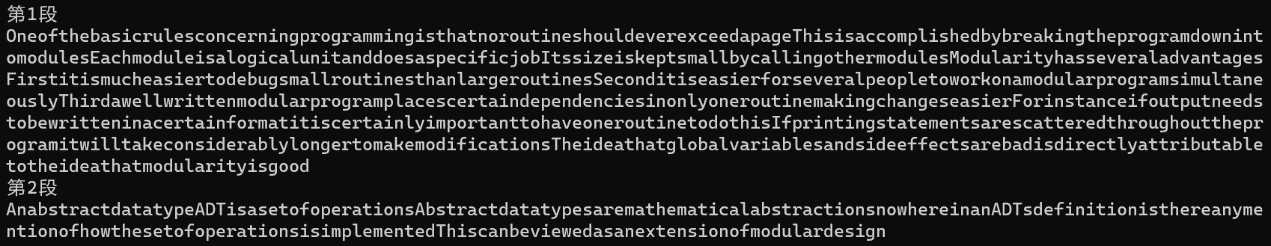
第2段

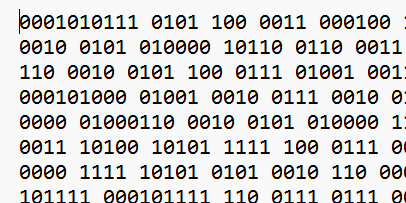
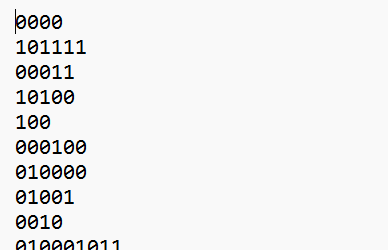
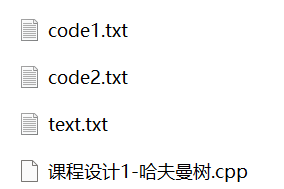
AnabstractdatatypeADTisasetofoperationsAbstractdatatypesaremathematicalabstractionsnowhereinanADTsdefinitionisthereanymentionofhowthesetofoperationsisimplementedThiscanbeviewedasanextensionofmodulardesign

时间复杂度：总体时间复杂度主要取决于对文章字符的处理，因此可以表示为 O(N)，其中 N 是文章的字符数。不过，需要注意哈夫曼编码表的构建和输出等操作也是常数时间操作，对总体时间复杂度的影响较小。

结果截屏：







文档 code1 code2

心得体会：学会了哈夫曼的建立，每次从未创建结点的结点中找最小的两个生成父亲结点，再把父亲结点的值放回数组中。再这个案例中发现光弄英文字符是不足以让解码后的文档可读的，还需要考虑空格，标点，数字等，给他们也要赋予哈夫曼编码才能更好的使用。对用什么样的数据结构有了深刻的思考。