

Coding Proj.3

This is a co-op homework from DCS211-Data Structure and Algorithm by Qiao Haiyan.

Build together with Haoqin Liu.

CO-OP CODING PROJECT REPORT:

Compress & Decompress File Using Huffman Tree

19335025 陈禹翰 chenyh369@mail2.sysu.edu.cn 19335137 刘皓青

2020年10月25日

摘要

本实验主要通过读取文件中的文本，通过构建哈夫曼树来压缩文件，并通过重建哈夫曼树来解压文件。

引言

问题：使用哈夫曼树来压缩和解压文件

哈夫曼树：又称为最优二叉树，是带权路径长度最短的树，权值较大的结点离根较近。

解决问题的思想：统计单词出现的频率，频率越高的权重越大，因而编码越短。

实现

main.cpp: top 调用

fileCompress.h: 压缩类

```
1  class handleFile {
2      HuffmanTreeNode *_chars[256];
3  public:
4      handleFile(); // 初始化节点数组
5      void myDFS(string &s, map<unsigned char, string> &m, HuffmanTreeNode
        *root); // 深搜
6      void recordWeight(map<unsigned char, string> &m, HuffmanTreeNode *root);
        // 记录频率
7      void toCompress(const char *filename, const char *outfilename, const char
        *antithesesName); // 压缩，生成压缩文件和一个频次表文件（非解压必需）
```

fileDecompress.h: 解压类

```

1 struct FILE_HEAD fileHead;
2     string str;
3     struct WEIGHT *w;
4     HuffmanTreeNode *_chars[256];
5     HuffmanTreeNode *root;
6 public:
7     void readFileHead(const char *sourceFileName, const char *outFileName);
8     void createHuffmanTree();
9     void decodeHuffmanTree();

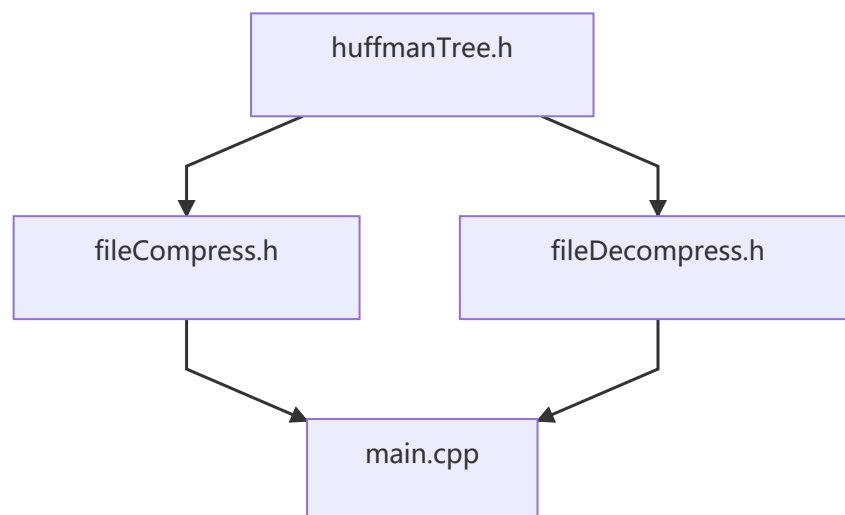
```

huffmanTree.h:

```

1 struct FILE_HEAD {                // 元数据，包括了字符种类和最后有效位。
2     unsigned char alphaVariety;
3     unsigned char lastValidBit;
4 };
5 struct WEIGHT {                   // 统计每个字符的频率
6     unsigned char _ch;
7     unsigned char _freq;
8 };
9 struct HuffmanTreeNode {         // HuffmanTreeNode
10     HuffmanTreeNode *_left;
11     HuffmanTreeNode *_right;
12     int _weight;
13     unsigned char _ch;
14 };
15 class cmp;                       // compare in priority_queue
16

```



程序使用和测试说明

使用 `g++` 编译

```
1 | g++ .\main.cpp -o mian.exe
```

运行

```
1 | .\main.exe argv[1] inputFile outputFile  
2 | argv[1] 可以是:encode,decode,compress,decompress.
```

调试时可以打开宏定义的 `DEBUG` 为 `true`

总结和讨论

学会了许多文件操作，也深刻了解了 `Huffman Tree` 的原理。过程中遇到了许多问题，昨晚调试到很晚发现bug是left和right写反。此次代码还有许多待完善的地方，其中刘皓青同学事先写了压缩类，我把频率整合进一个文件并实现了解压。事先做好设计才能减少写代码时的错误。并且与他人协作要养成良好的代码风格。

参考文献

https://baike.baidu.com/item/%E5%93%88%E5%A4%AB%E6%9B%BC%E6%A0%91/2305769?fr=aladdin#7_1