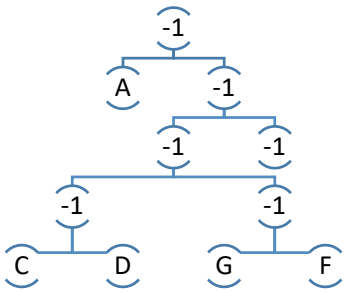
Instruction

Yuqing Zhong

1. Overview

My project is Huffman Compression, which consume a text file and return a coded or decoded file.

2. Changes
   - The structure of the program changes a lot, I use much more small functions to create a heap, build a Huffman tree, print the tree and print the coded file and etc.
   - Implement both encoding and decoding program in this project.
   - In encoding part
     - use a sum to count how many characters that have been encoded, which avoid decode redundant bits.
     - Use heap to create a Huffman tree.
     - Instead of writing a code chart to file, the program will write the tree to the file (as shown in `print_tree`). For example, the following tree will be coded as -1 65 -1 -1 -1 -1 -1 C D G F -2, where -2 shows the end of the tree code. By doing so, the program can rebuild a tree identical to previous one.



   - In decoding part
     - Use binary write to `fprintf` file, so that even special char like 1A(SUB) and 00(NULL) can be decoded.

3. How to Use it

```
> ./ Huff_Comp
Press 1 for encode, press 0 for decode:
1
Please enter the file name:
test.txt
Enter filename to write to:
coded.txt
```

To encode a file, enter 1 for encoding, and then enter the name of the file to compress. The compression will work with ANY file, like text file, xlsx file, pdf and etc..

```
> ./ Huff_Comp
Press 1 for encode, press 0 for decode:
0
Please enter the file name:
coded.txt
Enter filename to write to:
decoded.txt
```

To decode a file, enter 0 for decoding, and then enter the name of the file to decompress. The decompression will work with any text file that produced by the compress program.

4. Results
   - test.txt 426 B -> coded.huff 508 B
   - test.txt 17KB -> coded.huff 10KB
   - test.txt 64 KB -> coded.huff 38 KB
   - test.bmp 1590 KB -> coded.huff 241 KB
   - test.pdf 1964 KB -> coded.huff 1963 KB

   Basically, the differences between original file and coded file are comparatively big for big files versus small files. But the differences vary with file types. The compression program works better with files that contains more repetitions.

5. Lessons Learned
   - Use IDE instead of text editor!
   - Learned a new data type: heap
   - Use unsigned char to read binary file
   - Use binary mode to read and write files
   - A good method to serialize a binary tree

Thank you for the wonderful courses you gave this semester!