Assignment 3

Hyoungshick Kim

October 20, 2020

1 File compression

This is a programming assignment to test your understanding of the Huffman coding algorithm.

- Your goal is to implement a program that compresses and uncompresses files using Huffman coding.
- You will write a code in the C programming language to compress and uncompress files using Huffman coding. The input file is an ASCII text file. The details are as follows:
- To compress a file, your program will follow the following steps:
  - Read in the input file named ‘`hw3_input.txt`’, and calculate the frequencies of all alphabetic (a-z only) characters in the ASCII text file. Please ignore other characters except for alphabetic characters. Also, convert all uppercase characters into lowercase characters.
  - Construct a Huffman tree for all alphabetic (a-z only) characters that appear in the input file.
  - Output the header describing the tree to the output file named ‘`hw3_output1.txt`’. You can use any description to specify a binary tree.
  - Rewind the file to the beginning re-reading each symbol (character).
  - Append the (binary) encoding for each symbol to the output file ‘`hw3_output1.txt`’. Mark “`HEADEREND`” as the delimiter to separate the header from the compressed part.
- To uncompress the compressed file `hw3_output1.txt`, your program will follow the following steps:
  - Read in the Huffman tree from the compressed file `hw3_output1.txt`.
  - Use the header to reconstruct the Huffman tree.
  - Decode the input using the Huffman tree.
  - Write the binary code of each alphabetic character used in the input file and the decoded alphabetic (a-z only) characters to the output file named ‘`hw3_output2.txt`’. The binary codes should be sorted alphabetically.

- The following is an example of input and output files:

```
[Input file: hw3_input.txt]
huffman

[Output file: hw3_output1.txt]
(((a,m),f),((h,u),n))HEADEREND1001010101001000011

[Output file: hw3_output2.txt]
a:000
m:001
f:01
h:100
u:101
n:11
huffman
```

- You will be judged by (1) the correctness of compression and decompression results produced by your submitted program, (2) the file size of `hw3_output1.txt` and (3) the well-written document to explain your source code and the performance analysis of your algorithm. For test, we will use input files up to 1 megabytes in size. Also, 16GB RAM will be used for testing. Please test your code extensively with several inputs, so you are sure it works correctly.

- You cannot use any pre-defined algorithms and data structures except arrays provided by some libraries.

- Please upload your source code (c files), instructions to illustrate how your source code works, document to explain your code and the performance analysis to iCampus.

- **Your assignments must be your own original work.** We will use a tool to check for plagiarism in assignments.