Data Structure Homework #3
Comparisons between Huffman Encoding and Arithmetic Encoding

Ping-Chen,Chung(鍾秉辰)
Student ID: 110503007
Department of Communication Engineering
National Central University

January 9, 2023

Contents

1 Program

1.1 Execution

Please change to the project repository and type in

```
make && ./obj/main.o
```

to execute the analysis program.

1.2 Pseudocode for main program

```
generate 20 gibberish text.
for each text:
    encode using arithmetic encoding and print the elapsed time.
    decode using arithmetic encoding and print the elapsed time.
    encode using huffmann encoding and print the elapsed time.
    decode using huffmann encoding and print the elapsed time.
```

2 Analysis

Below is the Output of each program processing 20 gibberish text: From the table, we can see that the execution time of Huffman coding in our selected repository is longer, which breaks down to the sections we will discuss below.

#	Arith. encoding	Arith. decoding	Huff. encoding	Huff. decoding
1	0.000138	0.000402	0.000264	0.001167
2	0.000144	0.000134	0.000189	0.001261
3	0.000296	0.000120	0.000257	0.001261
4	0.000110	0.000274	0.000204	0.001187
5	0.000109	0.000167	0.000193	0.001190
6	0.000117	0.000137	0.000250	0.001237
7	0.000233	0.000333	0.000186	0.001431
8	0.000111	0.000278	0.000242	0.001448
9	0.000237	0.000140	0.000412	0.001342
10	0.000241	0.000283	0.000490	0.001317
11	0.000228	0.000145	0.000285	0.001266
12	0.000111	0.000278	0.000287	0.001603
13	0.000116	0.000298	0.000281	0.001161
14	0.000107	0.000286	0.000244	0.001496
15	0.000236	0.000288	0.000294	0.001551
16	0.000231	0.000114	0.000340	0.001314
17	0.000091	0.000107	0.000331	0.001679
18	0.000246	0.000104	0.000295	0.001391
19	0.000090	0.000131	0.000240	0.001122
20	0.000263	0.000268	0.000252	0.001237
Avg	0.000172	0.000206	0.000265	0.001333

Table 1: Caption

2.1 I/O File difference

In arithmetic encoding, the output of a string could be perceived as a binary float number, which represent each character of the string. The program only needs to parse each bit and restore the bits of the string based on encoded file.

However, in Huffman encoding/decoding, the program has to decode the file based on an external chart, which represents the encoded bits of each character. In this way, the program will take more time making file access, and thus slows the execution time of file parsing.

References

[1] https://par.cse.nsysu.edu.tw/ homework/algo01/8934609/index.html

[2] https://github.com/ludlows/chuffman