Assignment #3

110503507 林亞誼

1. 編譯結果

HUFFMAN

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assignment_
ment_3_opensource# cd Huffman-main
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assignment_
3_opensource/Huffman-main# gcc -o huffman huffcode.c huffman.c huffman.h
```

#figure1. compile Huffman

ARITHMETIC

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assignment_
ment_3_opensource# cd Arithmetic-main
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assignment_
3_opensource/Arithmetic-main# g++ -o Arithmetic-main main.cpp Arithmetic.cpp
```

#figure2. compile Arithmetic

2. 執行結果

TEST 1

Huffman-main > test1

```
1  Tech giant's new robotaxi with removable steering wheel not legal yet in China
2  Chinese tech giant Baidu revealed its plan to roll out Apollo RT6, an
   autonomous self-driving car with a detachable steering wheel, even though
   Chinese authorities haven't approved the concept yet.
3  In August, Baidu obtained the first permits to operate fully driverless
   robotaxis in two cities in China.
4  01:18 - Source: CNN Business
```

#figure3. the content of test1

RESULT 1 for Huffman

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assig
nment_3_opensource/Huffman-main# ./huffman -i test1 -o encodingtext -c
time cost of huffman coding: 0.001111 /s.
```

#figure4. The result of Huffman encoding

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assig
nment_3_opensource/Huffman-main# ./huffman -i encodingtext -o decodingtext -d
time cost of huffman coding: 0.000525 /s.
```

#figure5. The result of Huffman decoding

RESULT 1 for arithmetic

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assig
nment_3_opensource/Arithmetic-main# ./Arithmetic-main

壓縮總bit數3312
其中為0的數為1802
其中為1的數為1510

bit0出現的機率  0.5440821256
bit1出現的機率  0.4559178744

time cost of arithmeric encoding :0.0029340000/s

time cost of arithmeric decoding :0.0000980000/s
```

#figure6. The result of arithmetic encoding and decoding

TEST2

Huffman-main > test2

```
1  abcdef
```

#figure7. The content of test2

RESULT 2 for Huffman

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assig
nment_3_opensource/Huffman-main# ./huffman -i test2 -o encodingtext -c
time cost of huffman coding: 0.001030 /s.
```

#figure8. The result of Huffman encoding

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assig
nment_3_opensource/Huffman-main# ./huffman -i encodingtext -o decodingtext -d
time cost of huffman coding: 0.000396 /s.
```

#figure9. The result of Huffman decoding

Arithmetic-main > main.cpp > main()

```
13      clock_t start, end;
14      start = clock();
15      test.open("test2.txt");
16      test.Press();
17      end = clock();
18      cout << endl<<endl<<"壓縮用時";
19      cout << double (end - start)/CLOCKS_PER_SEC<< "/s" << endl << endl;
20      start = clock();
21      test.Decode("C:/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/
22      110503507_assignment_3_opensource/Arithmetic-main/test2.txt.ari",
23       "C:/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/
24       110503507_assignment_3_opensource/Arithmetic-main/test2.txt");
25      cout << "解壓用時" ;
26      cout << double (start - end)/CLOCKS_PER_SEC << "/s" << endl << endl;
27      getchar();
```

#figure10. change test1 into test2

RESULT 2 for arithmetic

```
root@DESKTOP-SIDL7N9:/mnt/c/Users/User/OneDrive - 國立中央大學/桌面/大二上hw/資結/110503507_assig
nment_3_opensource/Arithmetic-main# ./Arithmetic-main

壓縮總bit數48
其中為0的數為27
其中為1的數為21

bit0出現的機率  0.5625000000
bit1出現的機率  0.4375000000

time cost of arithmeric encoding :0.0009940000/s

time cost of arithmeric decoding :0.0001020000/s
```

#figure11. The result of arithmetic encoding and decoding

3. 編碼原理

HUFFMAN 是一種可變長的分組編碼，先對準被壓縮的資料進行機率分析，對於發生機率較高的資料，以較短的位元串(Bit Strings)表示;對於發生機率較低的資料，則以較長的位元串表示。二進制的 HUFFMAN 是基於二元樹的編碼思想，所有可能的輸入符號在 HUFFMAN 樹上對應一個節點，結點的位置就是該符號的 HUFFMAN 編碼。為了構造出唯一可譯碼，這些節點都是 HUFFMAN 樹上的終端結點（即葉子結點），不會出現前綴碼。

ARITHMETIC 為了克服 HUFFMAN 編碼的局限性，基於非分組的編碼方法因而產生。它實際的編譯碼過程比較複雜，但具有許多優點，特別是需要的參數很少，不像 HUFFMAN 需要一個很大的碼表。具體編碼方法：將文件讀入，並將其分割成 bit 串形式，計算文件中 bit0 和 bit1 的總數量和各自的概率，對一定長度 L 的符號串進行編碼，並將數據寫入壓縮後的文件，從壓縮文件中讀入數據，並還原成長度為 L 的符號串輸入至解壓文件中。

4.問題探討

經由開源程式了解到除了 Huffman 和 Arithmetic coding 還有 LZ-78 可以進行壓縮並解壓縮。但是執行 Three-methods-for-comparison_huffman_LZ78_arth 此資料夾時，不能正常執行，以至於無法從中看到應展現的結果。而 LZ 編碼是由以色列研究者跳脫哈夫曼碼和算術編碼的設計思路，設計出的一系列比哈夫曼編碼更有效、比算術編碼更快捷的通用壓縮算法。LZ 系列算法利用一種巧妙的方式將字典技術應用於通用數據壓縮領域。後續可繼續嘗試此編碼方式已進行深入的探討。

5.参考資料

(1) https://blog.csdn.net/weixin_43825245/article/details/86480495

(2) https://github.com/starf1ame/Three-methods-for-source-

(3) encoding/blob/master/%E7%AE%97%E6%95%B0%E7%BC%96%E7%A0%81/main.cpp

(4) GitHub - drichardson/huffman: huffman encoder/decoder

(5) https://edition.cnn.com/videos/media/2022/12/26/trump-reelection-campaign-nymag-

nuzzi-cnntm-bts-vpx.cnn