

## Huffman code

使用 Huffman code 編碼對檔案執行壓縮，所耗費的時間為 0.000857s

將被壓縮的檔案進行解壓縮，所耗費的時間為 0.000747s

(檔案內容為 IMPENGRULUNG)

```
pengru@ubuntu:~/109503510_assignment_4/huffman-main/huffman-main$ gcc -o huffmain huffcode.c huffman.c huffman.h
pengru@ubuntu:~/109503510_assignment_4/huffman-main/huffman-main$ ./huffmain -i try -o ttry -c
PI = 3.142172
Time = 0.000857
pengru@ubuntu:~/109503510_assignment_4/huffman-main/huffman-main$ ./huffmain -i ttry -o ttry -d
PI = 3.142172
Time = 0.000747
pengru@ubuntu:~/109503510_assignment_4/huffman-main/huffman-main$
```

將一本英文小說檔案大小為 1.7MB 的 txt 檔進行壓縮，壓縮的時間 0.187846s

解壓縮的時間為 0.168613s，壓縮完的檔案大小為 987.2kB

```
pengru@ubuntu:~/109503510_assignment_4/huffman-main/huffman-main$ ./huffmain -i book5.txt -o book5e -c
PI = 3.142172
Time = 0.187846
pengru@ubuntu:~/109503510_assignment_4/huffman-main/huffman-main$ ./huffmain -i book5e -o book5d -d
PI = 3.142172
Time = 0.168613
```

## Arithmetic code

先對檔案進行編譯(檔案內容為 IMPENGRULUNG)

```
pengru@ubuntu:~/109503510_assignment_4/arcd-master/arcd-master/examples$ gcc -o arcdmain arcd_stream.c adaptive_model.c adaptive_model.h arcd.h arcd.c
```

對檔案執行壓縮，所耗費的時間為 0.000057s

將被壓縮的檔案進行解壓縮，所耗費的時間為 0.000114s

```
pengru@ubuntu:~/109503510_assignment_4/arcd-master/arcd-master/examples$ ./arcdmain -e <try|tee tryo
Ig[***GyQ*y*
PI = 3.142172
Time = 0.000057
pengru@ubuntu:~/109503510_assignment_4/arcd-master/arcd-master/examples$ ./arcdmain -d <tryo|tee tryd
IMPENGRULUNG
PI = 3.142172
Time = 0.000114
```

將同一份 txt 檔進行壓縮，壓縮的時間為 2.304575s，壓縮完的檔案 980.3kB

```
pengru@ubuntu:~/109503510_assignment_4/arcd-master/arcd-master/examples$ ./arcdmain -e <book5.txt|tee book5e
```

```
PI = 3.142172
Time = 2.304575
```

解壓縮的時間為 3.716346s

```
pengru@ubuntu:~/109503510_assignment_4/arcd-master/arcd-master/examples$ ./arcdmain -d <book5e|tee book5d
```

```
PI = 3.142172
Time = 3.716346
```

## 分析

Huffman coding 是使用機率低編碼長、機率高編碼短的原理進行編譯，

arithmetic coding 是將一段 message 用 0 到 1 間的區間來表示。比起 Huffman

coding，arithmetic coding 壓縮完的檔案小了一點點，比較特別的是 arithmetic

coding 在檔案小的時候壓縮解壓縮的速度較快，但在檔案大的時候壓縮解壓縮

的速度比較慢，推測可能原因是因為 arithmetic coding 運算複雜度較高所造成的

結果。