

壓縮演算法 Final Report

110503513 通訊二 王家欣

一、Build guide

1. Arithmetic coding

```
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4$ cd arcd
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/arc$ make
[ 25%] Built target arcd
[ 50%] Built target adaptive_model
[ 75%] Built target arcd_stream
[100%] Built target codec_tests
○ cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/arc$
```

2. Huffman coding

```
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4$ cd huffman
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/huffman$ make
cc -g -Wall -Werror -Wextra -O0 -std=c11 -D_POSIX_C_SOURCE=2 -c -o huffcode.o huffcode.c
cc -g -Wall -Werror -Wextra -O0 -std=c11 -D_POSIX_C_SOURCE=2 -c -o huffman.o huffman.c
ar r libhuffman.a huffman.o
ar: creating libhuffman.a
cc -o huffcode huffcode.o libhuffman.a
○ cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/huffman$
```

二、Execute guide

1. Arithmetic coding – compress

```
○ cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/arc$ ./examples/arc_stream -e < hello.txt | tee hellocompress.txt
```

2. Arithmetic coding – decompress

```
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/arc$ ./examples/arc_stream -d < hellocompress.txt | tee hellocompress.txt
```

3. Huffman coding – compress

```
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4$ cd huffman
○ cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/huffman$ ./huffcode -i hello.txt -o hellocompress.txt -c
```

4. Huffman coding – decompress

```
● cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/huffman$ ./huffcode -i hellocompress.txt -o hellocompress.txt -c
```

三、執行結果

1. Arithmetic coding – compress

```
compress time : 5062 microsecond
cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/arcd$
```

2. Arithmetic coding – decompress

```
compress time : 10156 microsecond
cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/arcd$
```

3. Huffman coding – compress

```
compress time : 1998 microsecond
cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/huffman$
```

4. Huffman coding – decompress

```
compress time : 4688 microsecond
cindy0727@LAPTOP-SIEN7KHU:~/projects/HW4/huffman$
```

5. Arithmetic coding – 資料大小

hello.txt	2022/12/20 下午 11:24	文字文件	41 KB
hellocompress.txt	2023/1/10 上午 10:49	文字文件	30 KB
hellodecompress.txt	2023/1/10 上午 10:53	文字文件	41 KB

6. Huffman coding – 資料大小

hello.txt	2022/12/20 下午 11:24	文字文件	41 KB
hellocompress.txt	2023/1/10 上午 11:00	文字文件	30 KB
hellodecompress.txt	2023/1/10 上午 11:14	文字文件	41 KB

由以上壓縮和解壓縮 hello.txt 檔(壓縮和解壓縮同一筆資料)的時間可知，執行 Huffman coding 所運行的時間皆比執行 Arithmetic coding 來的快，但是就壓縮完後的大小相比，目前來看無明顯差別。Hello.txt 是我去唬爛產生器產生一篇名為 hello 的 15000 字文章，這個唬爛產生器產生的文章會一直出現冗言贅字，例如出現如果如果如果，就這兩種壓縮演算法而言，若是重複出現相同的詞彙且頻率很高，就有機會將資料壓縮得很小。

四、心得

Arcd 跑出執行檔真的超難用，上網查聽到的做法是說執行 **cmake** 後要建造 **build** 的資料夾並到裡面執行 **cmake ..**，後來詢問同學後，發現他們使用的是 **cmake .**，目前的我還不知道兩種的差別，待未來慢慢釐清。沒有 **fdopen** 函式需要透過新增 **#define _XOPEN_SOURCE 600** 和我查到的方法需要更改編譯器版本的辦法不同，無法 **found -larcd** 的指令需要執行 **ccmake**。是想都沒想到，幸好有同學的幫忙才順利用出執行檔，感謝大家出手相助。

