

Assignment_3

(1). 編譯結果

➤ Huffman

```
● casper@LAPTOP-1DKFTNB9:~/work/huffmancode$ 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
```

➤ Arithmetic

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd$ make
[ 25%] Built target arcd
[ 50%] Built target adaptive_model
[ 75%] Built target arcd_stream
[100%] Built target codec_tests
```

(2). 執行結果

➤ Huffman Compress

(1). 500 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/huffman$ ./huffcode -i testfile500.txt -o compressfile500.txt -c
Run Time: 364 μs
```

(2). 1000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/huffman$ ./huffcode -i testfile1000.txt -o compressfile1000.txt -c
Run Time: 433 μs
```

(3). 3000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/huffman$ ./huffcode -i testfile3000.txt -o compressfile3000.txt -c
Run Time: 609 μs
```

➤ Huffman Decompress

(1). 500 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/huffman$ ./huffcode -i compressfile500.txt -o decompressfile500.txt -d
Run Time: 224 μs
```

(2). 1000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/huffman$ ./huffcode -i compressfile1000.txt -o decompressfile1000.txt -d
Run Time: 295 μs
```

(3). 3000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/huffman$ ./huffcode -i compressfile3000.txt -o decompressfile3000.txt -d  
Run Time: 442 μs
```

➤ Arithmetic Compress

(1). 500 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd/examples$ ./arcd_stream -e <testfile500.txt | tee compressfile500.txt  
Run time: 120 μs
```

(2). 1000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd/examples$ ./arcd_stream -e <testfile1000.txt | tee compressfile1000.txt  
Run time: 214 μs
```

(3). 3000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd/examples$ ./arcd_stream -e <testfile3000.txt | tee compressfile3000.txt  
Run time: 593 μs
```

➤ Arithmetic Decompress

(1). 500 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd/examples$ ./arcd_stream -d <compressfile500.txt | tee decompressfile500.txt; clear  
Run time: 160 μs
```

(2). 1000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd/examples$ ./arcd_stream -d <compressfile1000.txt | tee decompressfile1000.txt  
Run time: 312 μs
```

(3). 3000 字元

```
● casper@LAPTOP-1DKFTNB9:~/work/arcd/examples$ ./arcd_stream -d <compressfile3000.txt | tee decompressfile3000.txt  
Run time: 922 μs
```

3. 分析

(1). 字元數越多，壓縮檔案所需的時間越長

(2). 運用 Arithmetic 演算法壓縮檔案所需的時間少於運用 Huffman 演算法

4. 參考資料

(1). [drichardson/huffman: huffman encoder/decoder \(github.com\)](https://github.com/drichardson/huffman)

(2). [wonder-mice/arcd: Simple arithmetic coding library in C \(github.com\)](https://github.com/wonder-mice/arcd)