資結第四次作業 開源程式的使用

編譯指令：

Huffman：
```
$ g++ -o test Compressor.cpp
```

Arithemtic：
```
$ make
```

執行結果：

起初我先分別用了一個單字、84 個單字、8657 個單字以及 120 個 a 的文件去測試兩種演算法所消耗的時間。之後再將 8657 個單字的文件，拆分成一個一個的單字，讓兩種演算法分別執行 1~8657 個單字的文件並比較兩者所消耗的時間。

Huffman：
```
./test empty.txt
```

一個單字：

```
Created compressed file: oneword.compressed
Compression is complete
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.001485 secs(base)
```

84 個單字：

```
Created compressed file: 84word.compressed
Compression is complete
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.001358 secs(base)
```

8657 個單字

```
Created compressed file: 8657word.compressed
Compression is complete
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.006591 secs(base)
```

120 個 a

```
Created compressed file: aword.compressed
Compression is complete
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.000914 secs(base)
```

Arithemtic：
```
./archive empty.txt
```

一個單字：

```
Encoding is done
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.000676 secs(base)
```

84 個單字：

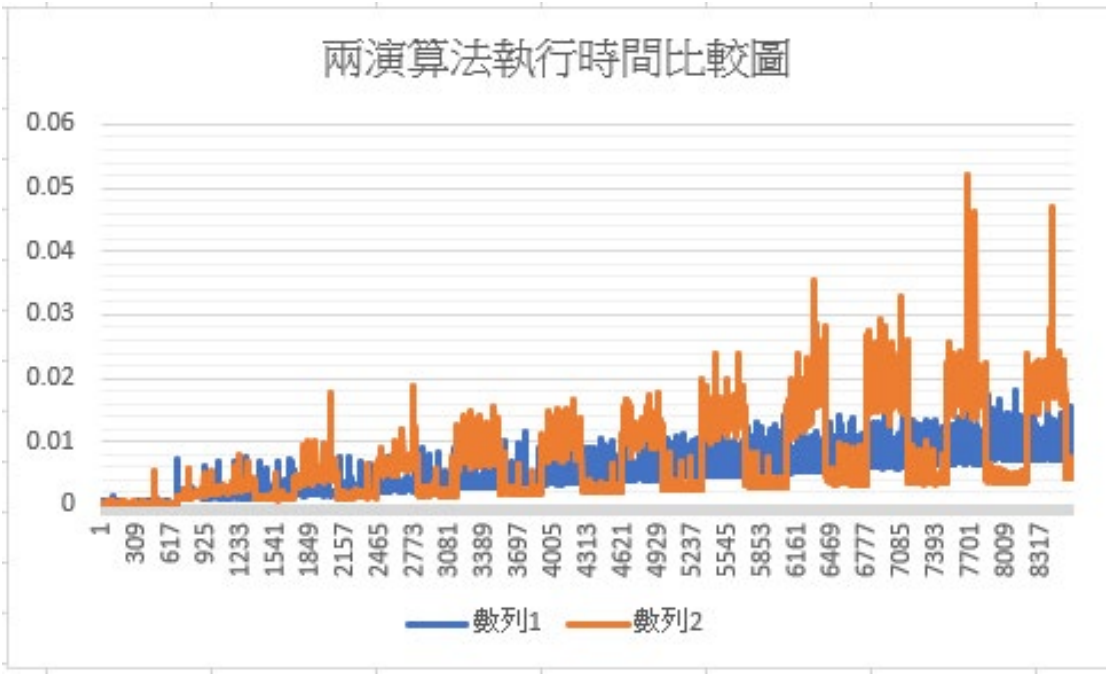
```
Encoding is done
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.000656 secs(base)
```

8657 個單字：

```
Encoding is done
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.012922 secs(base)
```

120 個 a

```
Encoding is done
constant CLOCKS_PER_SEC is: 1000000, time cost is: 0.000553 secs(base)
```



數列一為 Arithemtic，數列二為 Huffman

分析:

由圖可知 Arithemic 雖然在資料量小時較 Huffman 慢，但當資料超過約 1500 個單字之後 Arithemic 明顯比 Huffman 快速很多。在做圖之後，發現 Huffman 的時間非常奇怪，並非如 Arithemtic 一般穩定上升，而是時間有長有短，具體原因未知。