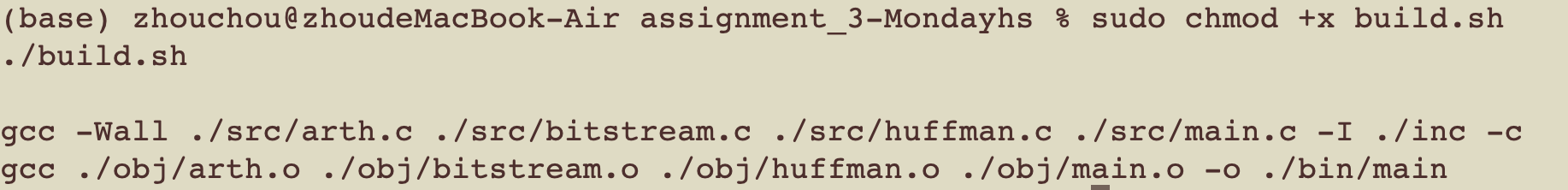
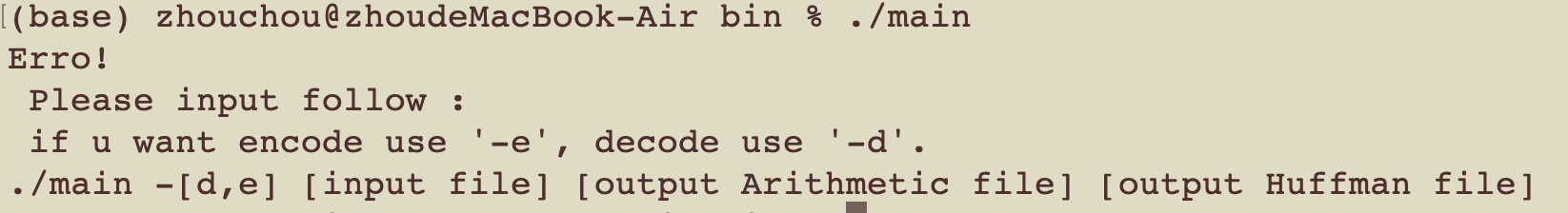
Assignment #3

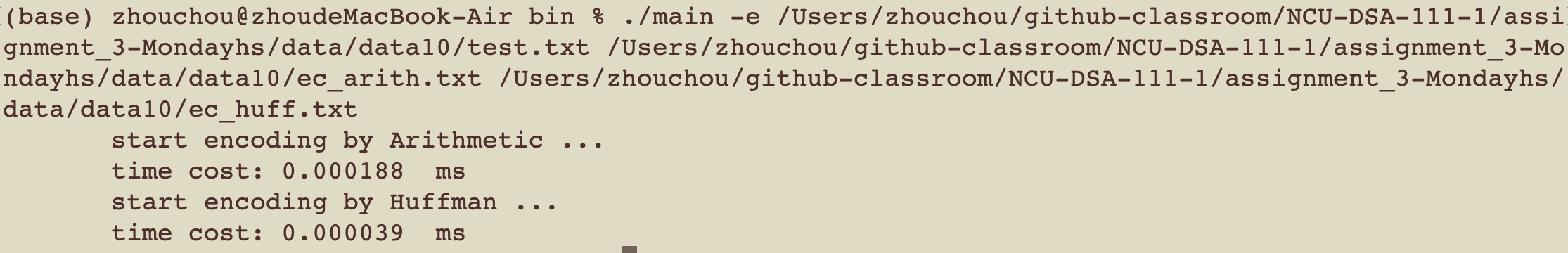
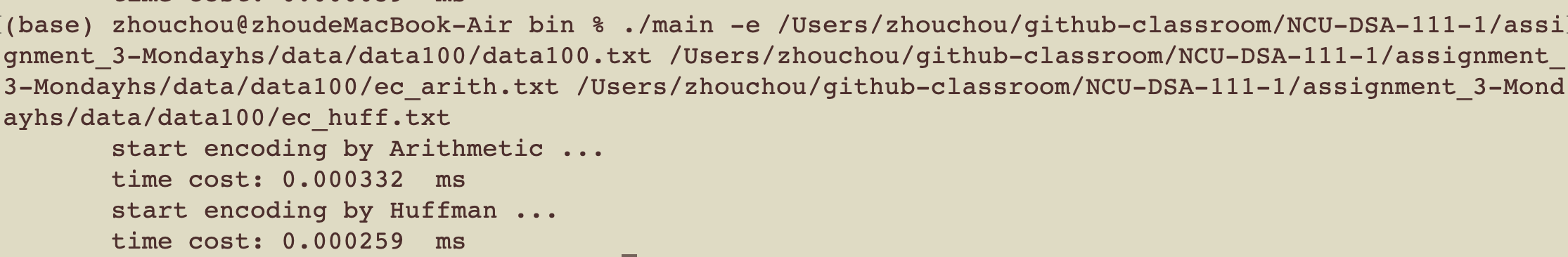
1. 編譯結果

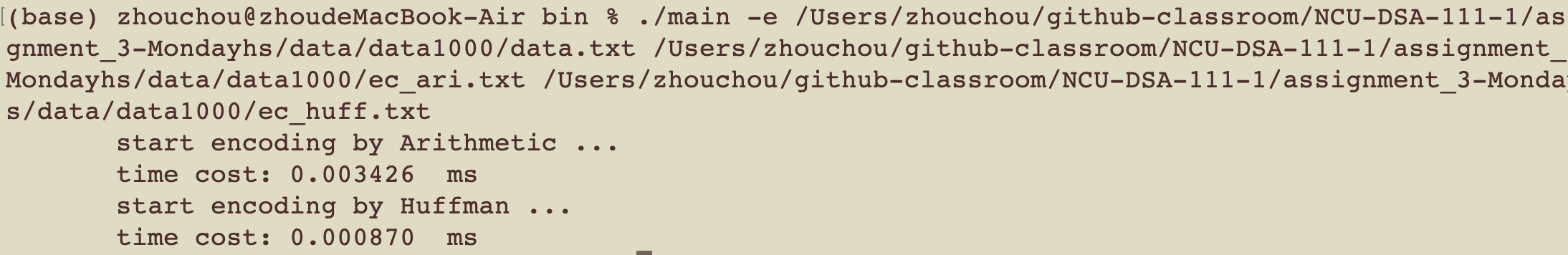


1. 執行結果



Error Run

Test data: 10 words

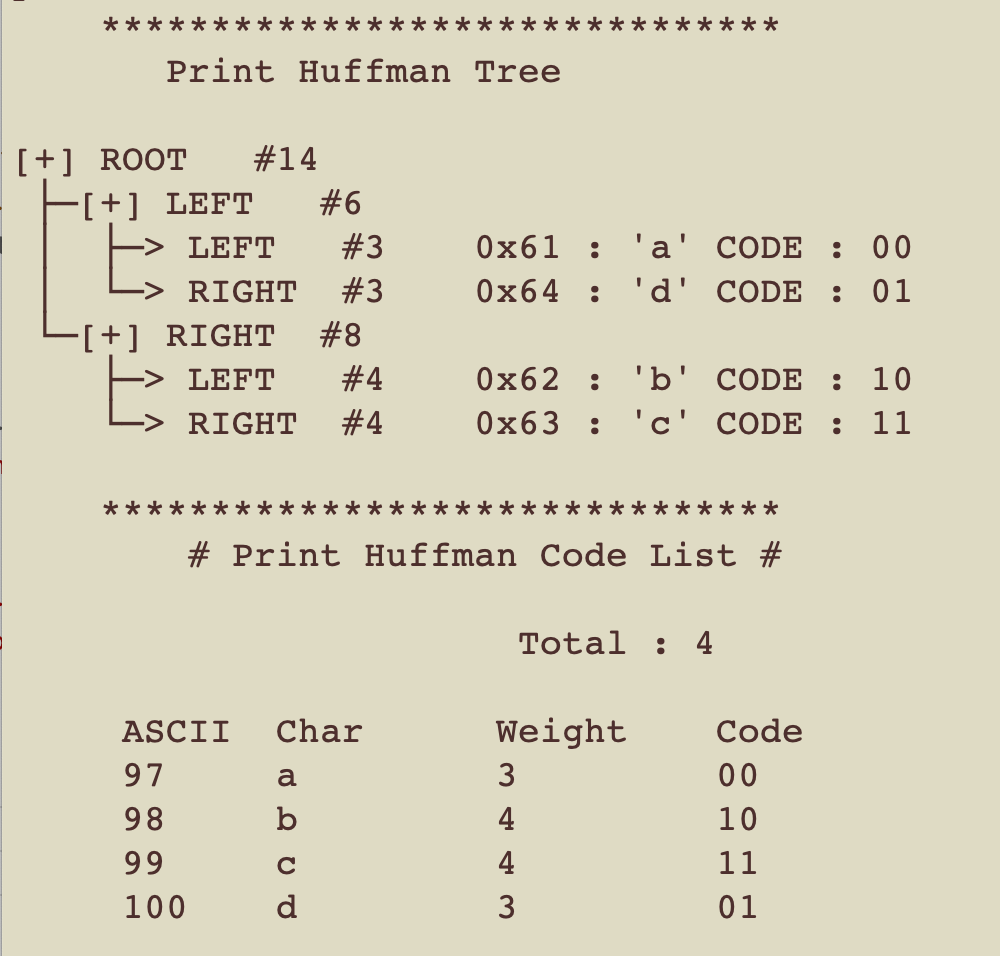
Test data: 100 words 

Test data: 1000 words

1. 原理

**Huffman:**

將欲壓縮之字串，先讀一遍，將字串中的每一相異單字元（Single Character）的出現頻率，做成統計，依此建構霍夫曼樹（Huffman’s Tree）。每一相異單字元，用0與1予以編碼，出現次數逾多者，給予較少的位元編碼。霍夫曼編碼法的特點在於所編碼出來的檔案具有唯一碼性質的即時碼。也就是各個相異字元所編碼出所位元串並不相同，解碼時能立即解出。



**Arithmetic:**

將一段message利用一個0到1間的區間來表示，當這個message越長，用來表示這個message的區間就越小，那要表示這個message的bit數就變多。相同的文字在message出現越多，區間變小的速度比較慢所以可以達到資料壓縮的效果。隨著符號序列長度之增加，其算數編碼長度之entropy趨近於無雜訊編碼理論之極限(效率愈高)。

1. 分析

當字數變多**Arithmetic**耗時就越長，**Huffman編碼時間較Arithmetic快很多。**

1. 未來

將兩者解碼的部分補上（能夠一起呈現）。

1. 參考資料

Huffman:

<https://www.foxzzz.com/Huffman-Code-Demo/>

Arithmetic: <https://par.cse.nsysu.edu.tw/~homework/algo01/8934609/index.html>