

Data Structure Assignment 3
Open Source的使用 Result Report

通訊二 110503518 李秉宸

January 10, 2023

1 Analysis 分析

1.1 時間比較

Table 1: 處理時間比較

文本	行數	字元數	Arcd Encode(s)	Arcd Decode(s)	Huff Encode(s)	Huff Decode(s)
1	1	59	0.000154	0.000282	0.000113	0.000044
2	2	112	0.000319	0.000463	0.000109	0.000044
3	4	179	0.000660	0.000906	0.000164	0.000064
4	9	574	0.001512	0.002263	0.000258	0.000167

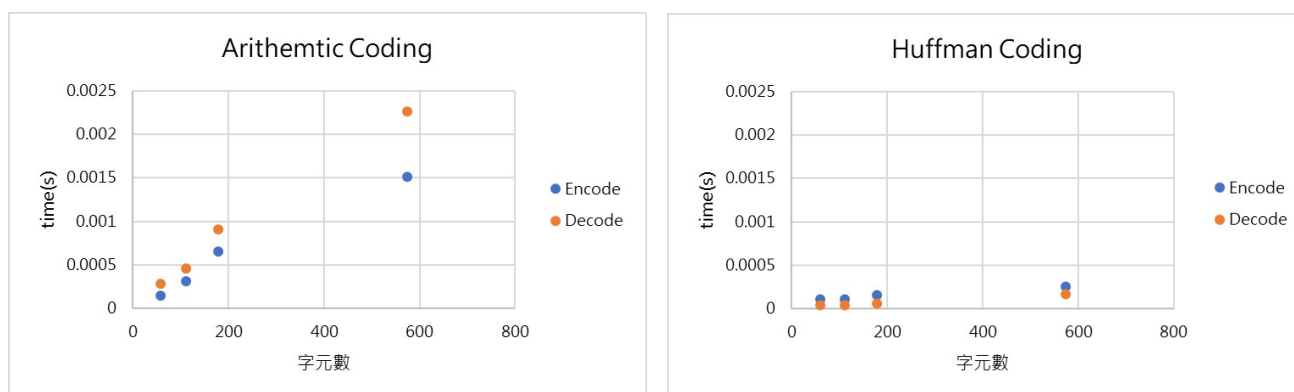


Figure 1: 圖表整理

由上圖與表可知，Arithemtic所花費的時間與字元數約成正比且增幅較為明顯，而Huffman花費時間則不因字元數而有顯著增加，整體而言，Huffman所花費的時間較Arithemtic少。文本容可見GitHub的test_text資料夾中。

2 Difficulties 遭遇問題

2.1 Arithemtic編譯問題

因未曾使用過cmake，加上原作者的include路徑或方法錯誤，我花了許多時間解決error，目前整理出兩種可行方法：

一、利用cmake .以及make編譯所有資料夾，但須手動修改include路徑、assert always True問題、fdopen/fileon問題(可利用define _XOPEN_SOURCE 600解決)、以及無法生執行檔的問題(可利用ccmake .將下圖ARCD EXAMPLES以及ARCD TESTS設為ON)。

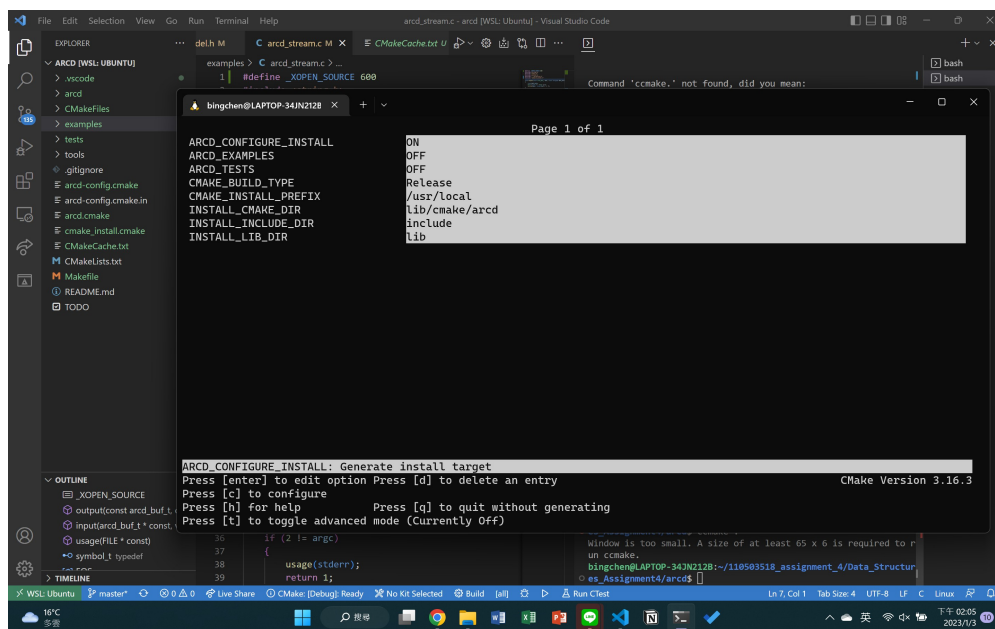


Figure 2: ccmake .

2.2 Arithemtic執行問題

經測試發現本程式有判定argc須為2，代表並不是用檔名視為引數的方式執行，而最後也與同學、助教討論後得知本程式讀取檔案的方式須使用bash指令。

2.3 抓取系統時間tick問題

上網找資料並參考他人作法後，我發現能利用time.h來抓取系統時間，也就是在程式運行開始與結束時分別抓取當下時間，並將兩者相減除以CLOCKS_PER_SEC，即可獲得系統運作時間(秒)。

3 Reference 參考資料

- [1] Arithmetic Coding [GitHub-wonder-mice/arcd:SimplearithmeticcodinglibraryinC](#)
- [2] Huffman Coding [GitHub-drichardson/huffman:huffmanencoder/decoder](#)