資料結構 final project report
電資院學士班 11006027 朱豐蔚

實作架構：

我會將整份 project 分成兩大部分進行講解，第一個部分是讀取資料夾中檔案轉換成 trie 存放，第二部分是讀取 query 進行搜尋的部分。

在已知道使用暴力搜索法在此專案中是不可行的，我便採取了助教在介紹影片中所使用的 trie 字典樹做為我此次專案中存放單字的主要結構。第一步驟是透過 argv 讀取資料夾，再藉由 file_open 加上 sprintf 加上 for 迴圈達到將每個 txt 檔案開啟的效果，並使用參考網路上的 trie class 架構，實作出了能夠將每個檔案讀取出來並將其內容的單字建成 trie 的專案架設。在這個部分中，我事先將所有英文字母轉換為小寫，以方便建樹與搜尋的大小寫混用。

在第二個部分，也就是讀取 query 中的搜尋單詞，我是採用 file_open 搭配 fout 來進行輸出與輸入的實作，而這部分最為困難的點在於如何完成專案中的各項不同的搜尋要求，像是 exact、prefix、suffix、and、or，我最後採用 getline 讀取每行的輸入，並搭配迴圈，從每行的頭開始判別，若是” 則進行 exact search、若是*則進行 suffix search，若是+則紀錄狀態為且，若是/則進行狀態為或，若為其他則進行 prefix search。注意，在這邊我也有將 getline 後的行進行全部轉換成小寫的轉換，已進行後續 trie 的搜尋。

我創立了最終搜尋的陣列與暫時搜尋的陣列，也就是說，每次的搜尋結果都會先放入暫時搜尋區，並根據當下狀態是且還是或，去與最終搜尋的陣列進行集合的更動，透過這樣，我能夠更省空間的，在記憶體不爆的狀態下完成這次專案。

最後結合第一與第二部分的專案，使用者可以在一開始給定目標的資料夾、輸入的 query 檔案、輸出的 txt 檔，實現搜尋論文單詞搜尋的專案，並將結果的論文標題存放於輸出的 txt 檔案當中。

遇到的困難與挑戰：

這次製作專案的過程中，其實我遇到蠻多挑戰的，第一個是 trie 新資料結構的熟悉與使用，要去認識 trie 背後的運算邏輯推理，並且將其加以應用，改造成能夠因應 prefix、suffix 的工具，這部分花了我蠻大的時間的。我透過創建新的 class function 將原先提供的 trie search 改造成能因應 prefix 的 trie prefix_search，使其能因應 prefix search 的需求。至於 suffix 的部分，我則是翻轉並建立 reverse string trie，來解決 trie 不能從尾巴搜尋的問題。

第二個挑戰，是在認識 txt 檔案的輸出與輸入，我們以往進行的 coding 都是在 oj 上，很少有像專案一樣，需要針對 txt 檔去做讀取與輸出，所以在 file reference 的參考上我查詢了蠻多網路上的資料，並加以精煉成適合用在專案當中的工具。

最後的挑戰則是在控制暫時搜尋與最終搜尋兩集合的交集與聯集，我在原先將其想得太過於困難，是採用 vector 來做交集與聯集，後來在進行 big data 的的搜尋時，就出現了嚴重 TLE 的情況。後來我回歸並回想在程設的所學，回歸最開始的陣列結構，就既簡單又快速的成功地解決了這個問題。這個改變也提醒了我，資料結構是各有好處的，就算我們在這堂課中到了多摩深奧的資料結構與演算法，都不該忘記最基本的資料結構陣列與 linked-list，能夠加以活用簡單的資料結構，並使用高階資料結構進行專案的加速，才是正確進行專案開發的過程。

參考資料：

1. 用 C++ 實作 Trie
   用 C++ 實作 Trie - CS Note (blueskyson.github.io)

2. C++ 檔案輸入和輸出 fout fin
   https://www.796t.com/content/1548759070.html

3. Cppreference
   cppreference.com

4. 助教的專案介紹影片