**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)](https://blueskyson.github.io/2021/09/15/trie/)

1. C++ 檔案輸入和輸出fout fin

<https://www.796t.com/content/1548759070.html>

1. Cppreference

[cppreference.com](https://en.cppreference.com/w/)

1. 助教的專案介紹影片