

- (i) 我的想法是把資料讀進來之後來判斷要創建幾個 `thread`，因此我是用動態陣列方法去儲存資料跟 `thread`，我每個 `thread` 的 `data` 有自己的搜尋字串、被搜尋字串、開始位置跟一開始 `data` 傳入的 `n`，然後用 `for` 迴圈讓他每個都 `join`，`thread` 會開始他們的工作，`thread` 會 `allocate` 一個空間存放他找到的答案然後傳位址回去，`main` 拿到他的位址後會用 `map` 包 `set` 的容器去儲存他，因為 `map` 可以用搜尋字串當 `key`，`set` 會自動排序比較方便，使用完後會釋放掉空間。最後把每個找到的位置輸出，我的設計是就算沒找到也會輸出那個搜尋字串，讓別人可以看到有幾個搜尋字串，之後用 `getrusage` 加一些計算取得他的 `user CPU time` 跟 `system CPU time`，把他們加起來，因為是秒所以要乘 1000 變成毫秒以符合題目規定，前面的兩個大的 `for` 迴圈是判斷加分題的?跟{}，更具體的我是在程式裡面寫註解。

(ii)

我都有完成基本要求跟進階要求

a

- i. 資料只有讀一次
- ii. 我的程式不限制 `n` 是 2
- iii. 每個 `thread` 都有傳遞他們的結果
- iv. `Thread` 是在自己的 `function` 裡用函式拿到自己的 `tid` 並

輸出

v. 有正確的印出 CPU 時間(ms)

b

- i. 我有判斷如果是?的話會替換成 A、T、C、G，因此假設原本是 2 個搜尋字串會變成有 5 個，create 的 thread 變成 $5*2=10$ 個，而且不限制只有一個?，可以有多個?
- ii. 我的程式有判斷如果是{}的話會裡面的替換然後放進去，因此假設原本有 2 個搜尋字串，其中一個是 AT{C,G}，它會變成 ATC 跟 ATG，因此搜尋字串會變成 3 個，create 的 thread 的會變成 $3*2=6$ 個，同樣的我也限制只有一個大括號可以有多個 a

我寫的程式不能判斷大括號裡面有問號，ex.{A,?}

(iii)

我的程式是這樣編譯的:

```
g++ -o 1083315_02 1083315_02.cpp -lpthread  
./1083315_02 data.txt
```

執行的時候後面加空格跟測試 data 的檔名

不知道為什麼有時候第一次跑程式格式容易跑掉，再重新執行一次格式就會對了。