

資結 練習二 報告

(一) 簡介:

首先讀檔並將每筆資料附上唯一序號後，確認原先有無建立過 23tree、avl tree 過，如有便先刪掉。

任務一：建立 2-3 樹

先進一個迴圈將前面讀檔進來的一筆一筆資料，先判斷 tree 是否為空再決定是要直接 create 一個節點還是用 insert 的進去做插入資料的動作，最後印出 2-3 樹的樹高以及樹跟的所有資料。

(1) tree 為空->先將這筆資料丟進一個 create 的 listing(資料庫)，再將此 listing

丟進一個 create 的節點的中間 listing 並同時設定其他東西(左小孩、中小孩、右小孩、父節點、最大值數量、最小值數量...)

(2) tree 不為空->insert 找到插入資料的位置

insert:首先先檢查此節點有沒有同樣的學校名，並用布林將是否找到的結果記錄

下來，如果有找到就會直接在此節點的那個 listing 數量加一並將動態陣列的序

號記錄下來(先前每筆資料有附上唯一序號)；如沒有便會先判斷這筆序號資料的

學校名是否小於目前此節點的中間值的 key，再判斷中間是否存在且比最小還大，

如有便(1)往中間走進行 insert 遞迴，如沒有則判斷能不能往左走(左小孩是否不

為空)，可以便(2)往左走進行 insert 遞迴，還是沒有則(3)creat 一個 Listing 並設

定節點中數字排行再找地方插入資料最後會調整節點；如這筆序號資料的學校名

大於目前此節點的中間值的 key，再判斷中間是否存在且比最大還小，如有便(1)往中間走進行 insert 遞迴，如沒有則判斷能不能往右走(右小孩是否不為空)，可以便(2)往右走進行 insert 遞迴，還是沒有則(3)creat 一個 Listing 並設定節點中數字排行再找地方插入資料最後會判斷有無須調整節點。

設定節點中數字排行:先判斷新資料是否大於中間值的 key，如是便判斷最大值是否存在(已經有兩個 listing)，再判斷新資料是否有比最大值還大，如果有則做交換位置(全部往後移)；沒有則新資料成為新的中間值，原中間值往後移；如果最大值不存在(目前只有中間值)，則直接將新資料放最大值。

如果新資料小於中間值的 key，如是便判斷最小值是否存在(已經有兩個 listing)，再判斷新資料是否有比最小值還小，如果有則做交換位置(全部往前移)；沒有則新資料成為新的中間值，原中間值往前移；如果最小值不存在(目前只有中間值)，則直接將新資料放最小值。

判斷有無須調整節點:首先檢查是否超過 3 個 listing，再判斷是否在樹頂，不在的話將中間值當作 newlist 往上丟，接著檢查父節點有無需調整節點，再來切割節點，新節點中各為各自的中間值，判斷父母是否有三個孩子，調整節點；在樹頂的話更動樹頂指標，切割節點，新節點中各為各自的中間值，設定新的首節點，將樹高加一，最後回傳新的最小值。

任務二：建立 AVL 樹

同樣，先進一個迴圈將資料一個一個推進 AVL 樹(insert)，最後印出 2-3 樹的樹高以及樹跟的所有資料。

Insert: 首先先建立新節點設定鍵值(有可能會有很多一樣科系名稱的，這時要將他們放在同一個節點)，再來設定每個節點中這個鍵值(科系名稱)所擁有的 data 數量為一，再將 newdata 推進這個節點的資料庫，將此新節點放進 insert 做遞迴尋找插入的位置。

Insert 遞迴: 首先判斷如果是第一個就直接放 root 並直接回傳目前節點；如果不是接著判斷新資料 key 是否比目前節點 key 大，如果比較大往右走做 insert 遞迴；判斷新資料 key 是否比目前節點 key 小，如果比較小往左走做 insert 遞迴；判斷新資料 key 是否等於目前節點 key，鍵值(科系名稱)一樣的話將那筆資料 allR 推進這個節點的資料庫，並將數量加一並且也是直接回傳目前節點。

接著加入完新資料後要檢查是否要做調整(從剛新增的一路檢查回來)，先算出此節點的平衡係數，再判斷平衡係數是否大於 1 且新節點的 key 是否小於目前節點左小孩的 key，如是則直接回傳目前節點往右轉而回傳的值；不是則接著判斷平衡係數是否大於 1 且新節點的 key 是否大於目前節點左小孩的 key，如是則更新並將目前節點的左小孩往左轉，回傳目前節點往右轉而回傳的值；不是則接著判斷平衡係數是否小於-1 且新節點的 key 是否大於目前節點右小孩的 key，如

是則回傳目前節點往左轉而回傳的值；不是則接著判斷平衡係數小於-1 且新節點的 key 是否小於目前節點右小孩的 key，如是則更新並將目前節點的右小孩往右轉，回傳目前節點往左轉而回傳的值，最後都不是則回傳目前的節點。

任務三：

首先，建立一個 target 節點，接著紀錄使用者輸入的學校名稱和科系名稱，判斷名稱中是否有*符號

1.學校名稱*符號與科系名稱:直接進任務二建立的 avl tree search 同樣科系名稱的所有資料(因為*代表求所有資料)

2.學校名稱與科系名稱*:直接進任務一建立的 23tree search 同樣學校名稱的所有資料(因為*代表求所有資料)

3.學校名稱與科系名稱皆為*: 直接將原本讀檔進來的動態陣列依序印出。

4.學校名稱與科系名稱: 將使用者輸入的學校名稱丟進 23tree search 紀錄所尋資料，再將使用者輸入的科系名稱丟進 avl tree search 紀錄所尋資料，最後將兩項資料庫進行比對，相同的便直接印出。

心得:

由於老師在這堂課給了我們警告說這次的作業有相當程度的難度，因此我們在前一次作業結束後立馬開始了作業二的準備與分配，包括看老師上課影片、兩個任務的撰寫和流程圖，由於任務一我們遇到了非常大的障礙卡了非常久，因此決定先將任務二完成，以免得不償失，而任務二較為簡單因此基本上是沒遇到甚麼問

題便成功完成。

任務一的部分，在建立 23 樹移動節點的部分，容易發生節點接點丟失，要一個一個把建樹過程畫出來才比較好容易解決這個問題。

任務三的部分我們猜測了很多有想過有可能是交叉搜尋(但覺得可能沒那麼簡單)、23 樹、紅黑樹...，因此我們有猜寫了多種版本，不過最後題目出來挑戰題是搜尋，雖然 23 樹、紅黑樹沒完全完成也沒派上用場不過也讓我們更加了解了老師所教的各種樹。

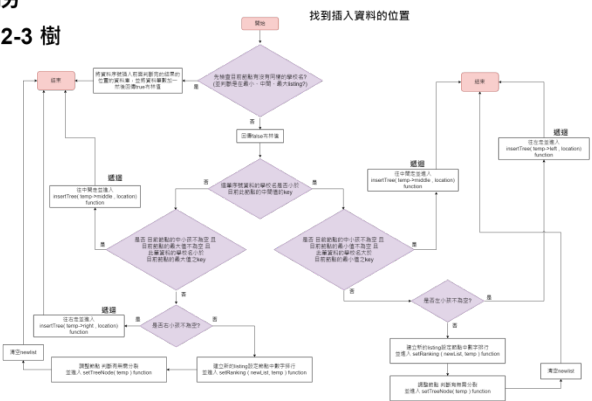
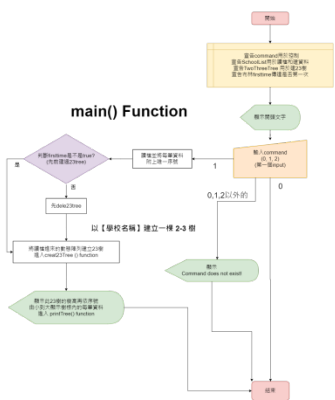
(二) 圖示:

任務一:

任務一
建立 2-3 樹

void insertTree(node* temp ,int location) Function

找到插入資料的位置

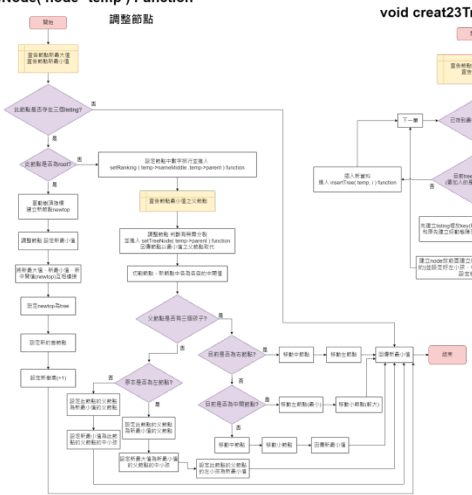
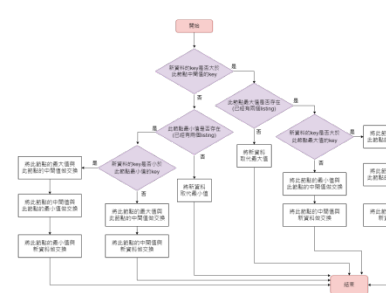


node* setTreeNode(node* temp) Function

調整節點

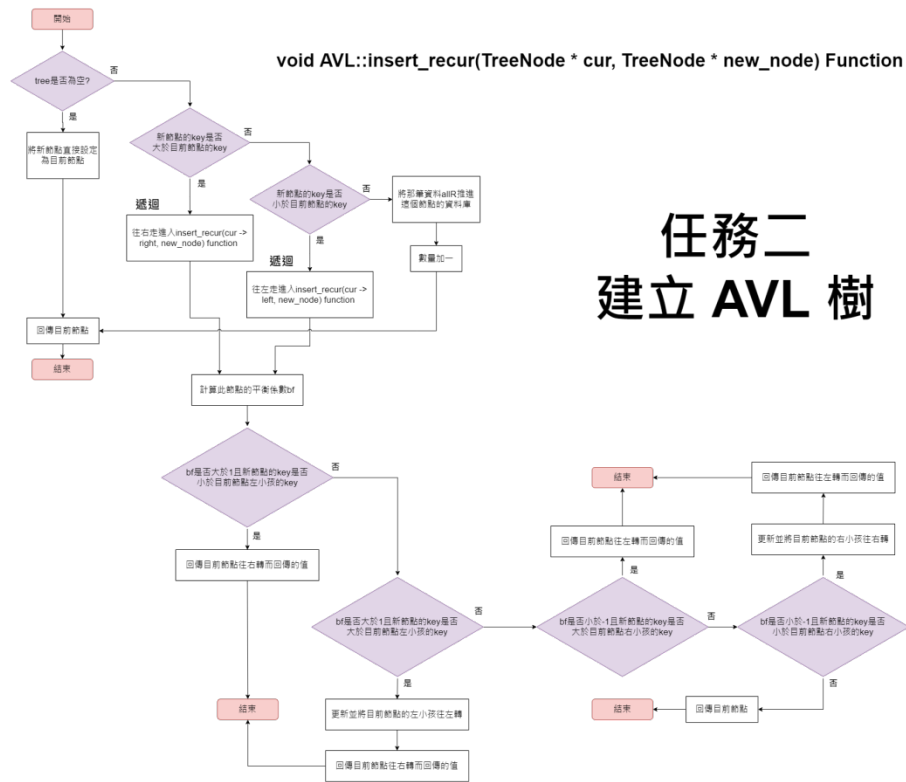
void creat23Tree () Function

設定節點中數字排行
void setRanking (listing& newList, node* temp) Function

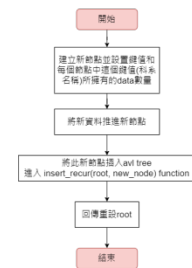


任務二:

任務二 建立 AVL 樹

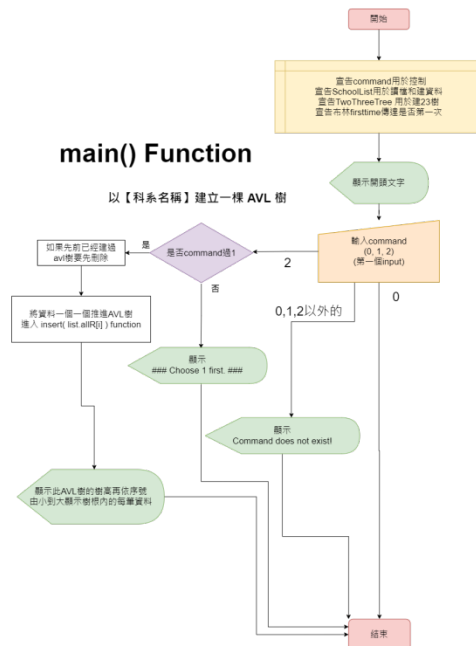


**void AVL::insert(SchoolList::schoolType newdata)
Function**
將資料一個一個插進AVL樹



main() Function

以【科系名稱】建立一棵 AVL 樹



(三) 答問

	Data203(622 筆)	Data204(2503 筆)	Data205(6407 筆)
任務一	96 byte	1304 byte	5440 byte

任務二	856 byte	5224 byte	10184 byte
-----	----------	-----------	------------