大家好我們是第13組，這次我們要介紹的是任務二的流程圖，首先我們讀input，如果沒有的話就繼續直到有為止，當我們開始讀檔時，我們使用迴圈跟linked list把資料存起來並把讀到的資料一筆一筆的傳入avl tree的insertnode來開始建樹。

avl tree的建樹跟二元搜索樹很像，都使用遞迴去做搜索且都用key value的大小來看此key的流向，舉例來說如果走到空的位置就建立新節點，有節點的話就比大小，小的話往左子遞迴，大的話往右子遞迴，而相同的話就存入把此節點的流水號vector裡，存完後我們會去看節點的BF平衡係數來去看需不需要旋轉，(BF的計算是左子樹高減去右子樹高)，當BF大於等於2時，就要做旋轉，至於是LL還是LR是看此node的左子的BF值，若BF小於0就做LL旋轉反之就做LR旋轉。

至於怎麼旋轉的話用LL旋轉為例子，把當前節點x和他的左子y做接線，把x的左子接向y的右子，y的右子接向x，最後回傳y就可拿到旋轉完的樣子。

而RR旋轉就是把LL的左右做相反，LR旋轉的話是他是先讓他的左子做RR旋轉，再讓本體做LL轉，基本上概念是差不多的。

而BF小於等於負2時也要做RL或RR旋轉，做RL和RR基本上跟LL和LR相似，小於改成大於，左右相反就好了。

至於BF在這區間的的話就什麼事都不用做。

由於有遞迴的特性，在我們insert完後我們會一直遞迴上去，從insert完的位置向上經過父節點一直到剛開始root檢查要不要旋轉，所以在這樣的檢查BF值下，我們不用擔心會有該旋轉的地方沒旋轉到。

--------------------

讀完資料建好樹後，我們開始算樹高和節點數，這點我們也是用遞迴就可以輕易地算出，算完後我們把根的資料做排序後回傳印出所有的資料，即可結束任務二。

換頁

而任務一的23樹，是比較複雜的題目，我們需要用vector來存三個指標，讀到資料用searchPath走到要走的位置，並用stack紀錄路徑，並把資料放到該放的位置，如果放完後葉子只有兩個item或是相同的值的情況那就結束了。如果那個葉有三個item就要做分裂，用stack走向父節點把分裂的中間值提上去，並且記錄指標，如果堤上去後還是滿的就繼續分裂到不需要為止，並記錄此時分裂的指標，這種情況下可能會需要紀錄4個指標當temp，如果分裂到根時候，需要創造新的節點，且分裂完後需要重新記錄根的位置，紀錄完後即可結束。

而這次的發現是我覺得這兩題的難度有蠻大的差距，AVL由於之前寫個二元搜索樹所以很輕易可以寫出，而2-3樹的難度是跟不同二元樹不同，他是有可能有三個指標的，麻煩的部分是分裂，分裂時要考慮它是葉子還是不是葉子，而且分裂的話還要考慮指標的流向，如果沒有注意到的話，很容易指錯的造成錯誤，更糟的話，會像寫組語那樣一直Segmentation fault ，這樣要debug時，會很難抓到bug。

這次的任務介紹到此結束，謝謝大家。