資結 練習一 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 heap、deap、min-maxheap 過，如有便先 reset。

任務一 ： 建立最大堆積 max heap

先進入一個迴圈依照序號由小到大一筆一筆新增至以學生數建立的 maxheap，再每次的資料更新後便重整一次 heap，從資料的尾端(剛新增的資料)開始檢查將較大的學生數資料往上推，最後印出最大堆積的樹根，底部節點，以及最左下角落的節點。

Reheap 的部分:設定變數的起始為資料的尾端然後去跑(HeapRebuid 的演算法)，先判斷此節點是否有 left child，有的話再判斷此 left child 的學生數是否大於最大值，如果有就將 left child 的學生數設為最大值，並將此子節點設定為要交換的位置，再來判斷 right child，方式跟 left child 一樣，如果節點有 right child 且學生數大於最大值，就設為最大值然後與要交換的位置做交換。最後判斷最大值(child 的學生數)是否大於自己的學生數，有就照之前設定好的位置將他們做交換，之後依照格式將 Heap 一行行印出。樹高的方面我們用 ceil()函數與樹高的性質 log(n+1)找出。最後要找最左邊與底部的節點，找最左邊節點的方法是如果目前節點還有 left child 就往左走，一直走到沒有，找到後依照格式印出此節點資料。若要找底部的節點就直接帶入 Heap 的總長度-1 的數，找到後依照

格式印出。

任務二 ： 建立雙堆積 DEAP

先建立一個虛擬的節點(0)，推進空的資料，再進入一個迴圈依照序號由小到大一筆一筆新增至以學生數建立的 DEAP，再每次的資料更新後便重整一次 DEAP，最後印出最大堆積的底部節點，以及最左下角落的節點。

ReDEAP 的部分:先算出序號 i 的 level，再藉由 level 算出 min 跟對應的 max 之間的差距，接著判斷序號 i 是在 min 還是 max，這部分我是算出 check(每 level 的第一個 max)，如果 i 有小於 check 代表在 min，反之再 max，接著

(1)在 min heap:算出 i 對應的 j 值(max)，檢查對應的 max 是否是空的，是的話就跟 j 的父節點比，反之就直接跟 j 比，如果 i 的學生數較大便交換且將 j 序列的資料與該父節點比較並作 maxheap 調整，反之則將 i 序列的資料與該父節點比較並作 minheap 調整

(2)在 max heap: 算出 j 對應的 i 值(min)，檢查對應的 min 是否是空的，是的話就跟 i 的父節點比，反之就直接跟 i 比，如果 j 的學生數較小便交換且將 i 序列的資料與該父節點比較並作 minheap 調整，反之則將 j 序列的資料與該父節點比較並作 maxheap 調整

任務三 ： 建立最小最大堆積 min-max heap

先進入一個迴圈依照序號由小到大一筆一筆新增至以學生數建立的 min-max heap，再每次的資料更新後便重整一次 min-max heap，最後印出最大堆積的

樹根，底部節點，以及最左下角落的節點。

Remin-max heap 的部分:先算出序號 i 的 level，接著判斷序號 i 是在 min 還是 max，這部分我是先判斷 level 是奇數還是偶數，如是奇數便在 min，反之在 max，接著

(1)在 min heap:比較序列 i 與之父節點的學生數，如果 i 的學生數較大便交換且將父節點與祖父節點比較將較大的往上推，反之則將目前節點節點與祖父節點比較將較小的往上推。

(2)在 max heap: 比較序列 i 與之父節點的學生數，如果 i 的學生數較小便交換且將父節點與祖父節點比較將較小的往上推，反之則將目前節點與祖父節點比較將較大的往上推。

心得:

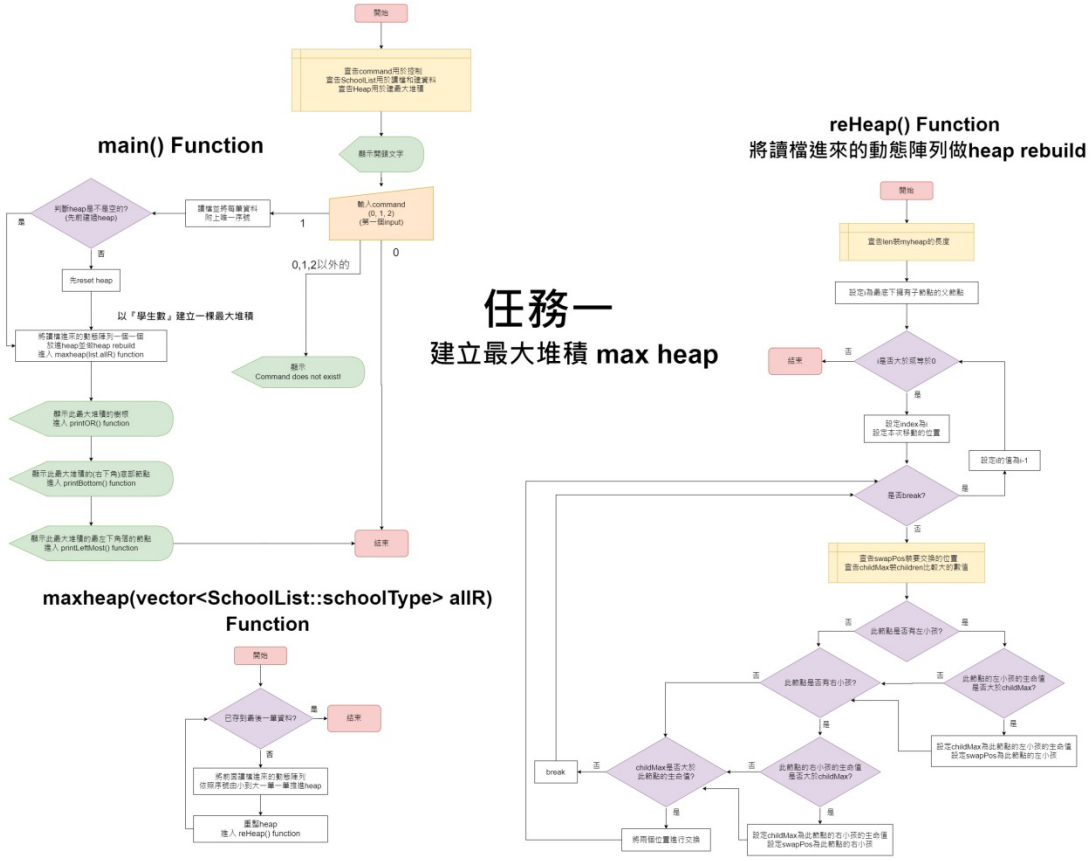
由於是剛開學的第一項作業，擁有相當充足的時間將老師的影片從頭到尾認真地看完，因此這次寫的作業一是完成的非常順利過程幾乎是沒有太大的障礙，不過在任務二的 deap 的部分由於我是用 vector 寫資料結構，因此我原本有煩惱了一段時間要如何假裝有一個空的節點，因為 vector 是用 push 進去無法在指定位置放資料，想的相當複雜後來才發現其實就放空的資料建立虛擬的節點(0)便能完成，覺得自己有時寫程式就是想的太多其實很簡單的方法便能完成。

在中間任務二的 deap 的部分我在判斷 min 還是 max 的部分卡關卡了比較久，一直想不出有甚麼辦法能判斷，甚至想說先寫任務三(有先猜好應該會是

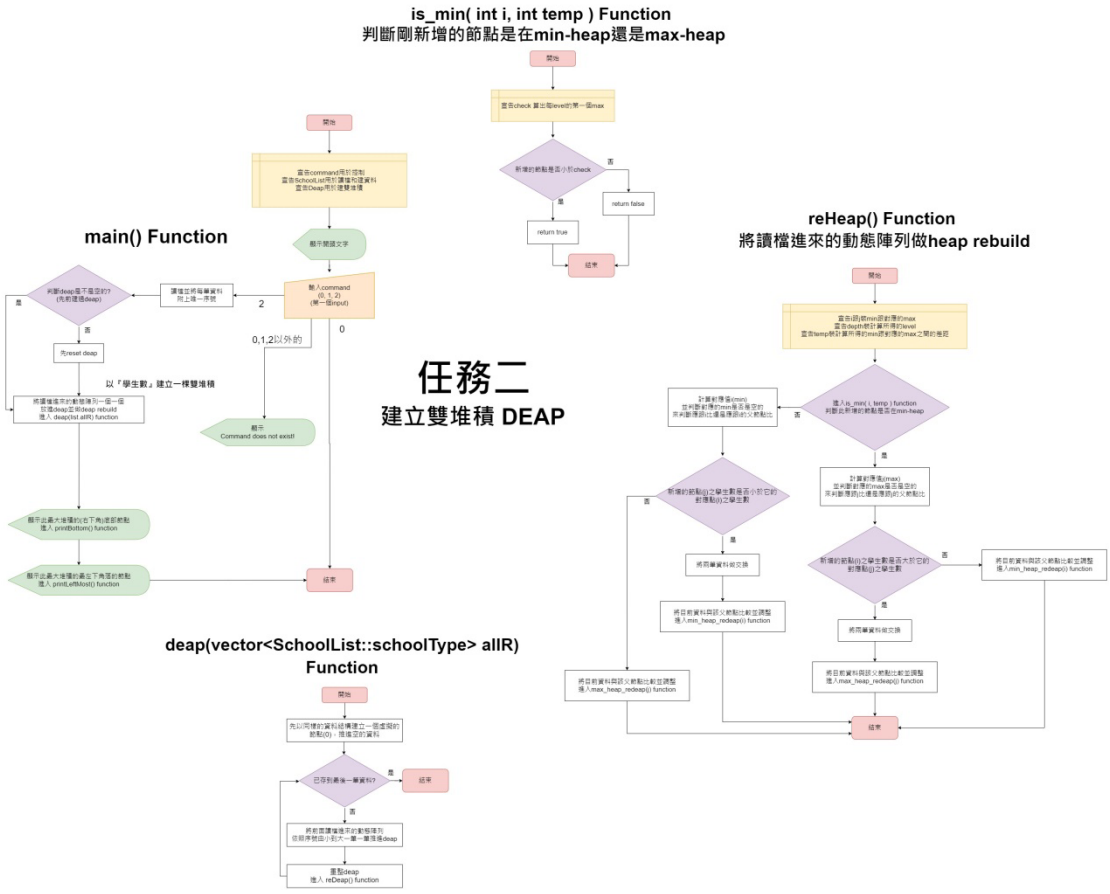
min-max heap 或是 delete)好了，結果在寫任務三的過程，仔細地又看過了老師的影片，後來便順利解決 deap 的判斷問題，真是太奇妙了，可能是因為這兩種 heap 其實差不多，於是讓我頓時豁然開朗，找到解決的辦法!!

(二) 圖示:

任務一:



任務二:



(三) 答問

	Data106(1040 筆)	Data104(2503 筆)	Data105(10073 筆)
任務一	17.2 ms	26.5 ms	448.4 ms
任務二	24.6 ms	31.9 ms	917.1 ms