第15組 10727124劉宇廷 10727139蘇崇傑

一、簡介

．main:

輸入可能的指令，若指令為0，程式執行結束；若指令為1，讀檔完後執行任務一；若指令為2，讀檔完後執行任務二；如果都不是，請使用者再輸入一次指令。

．任務一:

將list中資料分別複製到bubblesort及selectsort，再依序執行氣泡排序及選擇排序，最後分別計算排序時間，並顯示出來。

>>Bubble Sort

比較相鄰元素的畢業生人數，如果第一個比第二個小，就交換他們兩個的內容，從開始第一對到結尾最後一對，每一對相鄰元素重複同樣的動作，此時最後元素的畢業人數會是最小的數，持續對越來越少的元素重複上面的步驟，直到沒有任何一對數字需要比較。

程式中利用一個boolean值去記錄當前是否有執行交換操作，若沒有交換操作被執行，表示該數列已排序完成，即可直接跳出迴圈。

>>Select Sort

從未排序的數列中找到畢業生人數最大的元素，並將此元素的內容與已排序數列的最後一個元素交換，重複以上動作直到未排序數列全部處理完成。

概念是從i = 0到 n-2 ，作 n-1 回合，每回合自第 i+1筆到第 n-1 筆中找出最大的畢業人數，與第 i 筆資料做交換。

>>Quick Sort

在數列中任意挑選一個數(pivot)，然後調整數列使得所有在pivot右邊的元素的畢業人數都比pivot的畢業人數還小，而在pivot左邊的畢業人數都比pivot大的畢業人數，接著將所有在pivot左邊的數視為新的數列，所有在pivot右邊的數視為另一個新的數列，分別重複上述步驟(選pivot、調整數列)，直到分不出新的數列為止。

．任務二:

將list中資料分別複製到mergesort及quicksort，再依序執行合併排序及快速排序，最後分別計算排序時間，並顯示出來。

>>Merge Sort

分為兩大部分：拆分與合併。

拆分是把數列切一半成為兩個小數列，再把切好的兩個小數列再各自切一半，重複步驟直到每個小數列都只剩一個元素。

合併是排序兩個只剩一個元素的小數列，並且合併，再把兩邊排序好的小數列合併並排序，重複步驟直到所有小數列都合併成一個大數列。

．心得與發現:

本次作業最主要的問題是讀檔時發生的錯誤，一開始的想法是把資料一個一個讀入，分別讀進學校代碼、學校名稱、科系代碼、科系名稱...等，但是老師的規定中並無明確表示資料一定存在，只保證各個資料以tab作間隔而已，因此後來改成讀進一整行，再藉由計算tab個數找到畢業人數所在處，並儲存在對應的變數(graduaters)中。

二、流程圖

三、提問

Select Bubble Merge Quick Radix

1. 12 46 14 2 1 (800)

2. 182 981 59 10 8 (3139筆)

3. 2572 11738 342 94 25 (10075筆)

預期想法為:基數>快速>合併>選擇>氣泡，結果跟我們預期差不多，只是有些好奇氣泡排序優化過後的效率並沒有比優化前更好，原因可能是數列在排序前資料並無太大的可能是排序完成的，因此提早跳出迴圈的機會不高，在時間上的表現也沒有太大的突破。

選擇及氣泡排序的演算法都比較直接而簡單，當資料的數目很小時可以選擇使用，但當資料數目很大時，採用像是基數、快速及合併排序會是較聰明的選擇。