書面報告

一、開發環境

設備：Macbook pro m1

程式編輯器：VSCode

程式：Python（3.11.8）

二、實作方法和流程

相關實作工具

・使用 NumPy 做資料的切分，提高速度和效率

・用 time 函式庫的 performance counter 做計時

・用 multiprocessing 和 threading 函式庫實作 process 和 thread

・用 queue 作為 processes 間或 threads 間的數據共享與同步工具

各方法大致流程

方法一：一個資料列整個直接做氣泡排序

方法二：先把資料列做切分，將切分後的小資料列一個個丟到氣泡排序函式做排序，排序好的小資料列照順序放到 queue 中，都排好序後將兩兩丟給合併排序函式，做到直到所有小資料列合併成一個資料列

方法三：將資料列做切分並建立一個共享 queue，切分後的小資料列一個個丟到氣泡排序函式，將每個都設置成一個 process，等到每個氣泡排序的 process 都做完後把兩兩丟給合併排序函式，把每次合併都設置成一個 process，最後做一個等待，讓所有合併 process 都做完

方法四：將資料列做切分，切分後的小資料列一個個丟到氣泡排序函式，將每個都設置成一個 thread，等到每個氣泡排序的 thread 都做完後把兩兩丟給合併排序函式，把每次合併都設置成一個 thread，最後做一個等待，讓所有合併 thread 都做完

三、探討結果和原因

單位：ms（四捨五入到第二位）

K = {5, 10}	N = 1 w	N = 10 w
method 1	2526.27	258770.57
method 2	558.08, 277.79	55771.68, 28086.32
method 3	479.75, 558.50	15006.89, 8010.80
method 4	552.64, 290.60	56052.75, 27724.00

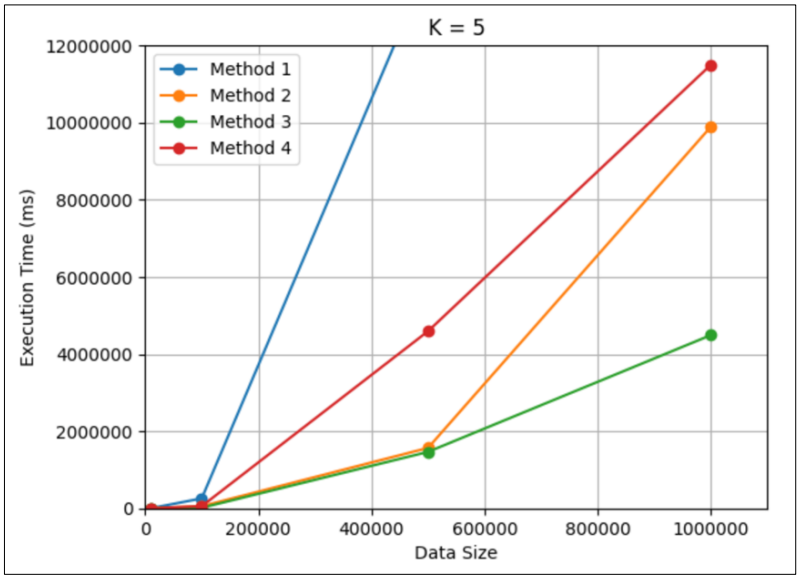
K = {5, 10}	N = 50 w	N = 100 w
method 1	14063328.60	50947088.54
method 2	1567509.01, 2044199.96	9893489.79, 3137331.38
method 3	1464121.38, 643341.19	4490925.84, 1603182.76
method 4	4592532.96, 1762576.45	11497888.77, 3063986.69

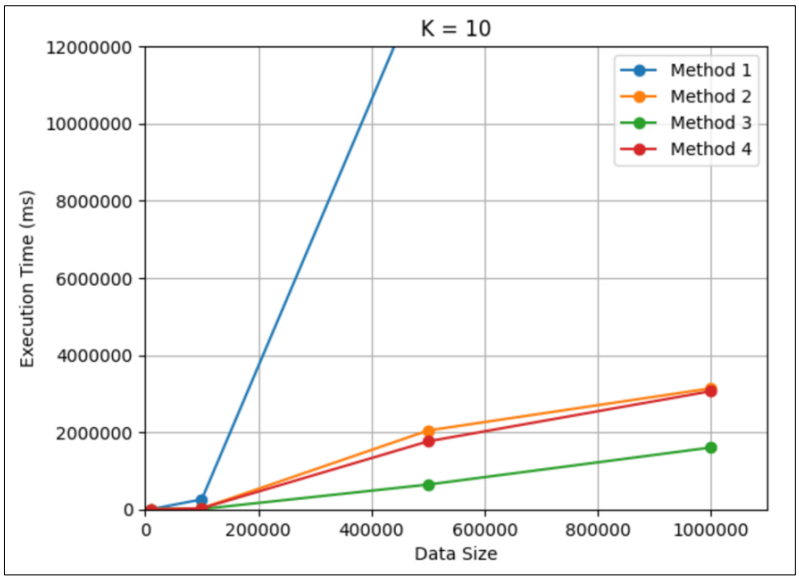
表一

N = {1w, 10w, 50w, 100w}	K = 5
method 1	2526.27, 258770.57, 14063328.60, 50947088.54
method 2	558.08, 55771.68, 1567509.01, 9893489.79
method 3	479.75, 15006.89, 1464121.38, 4490925.84
method 4	552.64, 56052.75, 4592532.96, 11497888.77

N = {1w, 10w, 50w, 100w}	K = 10
method 1	2526.27, 258770.57, 14063328.60, 50947088.54
method 2	277.79, 28086.32, 2044199.96, 3137331.38
method 3	558.50, 8010.80, 643341.19, 1603182.76
method 4	290.60, 27724.00, 1762576.45, 3063986.69

表二





由表一可知，方法一無論在什麼狀況都是最慢的，方法三則相反，在各種狀況時都是最快的，而在資料量為一萬時，K 是 5 時，方法四比方法二快；K 是 10 時，方法二比方法四快。其他無論資料量，在 K 是 5 時，方法二都比方法四快；K 是 10 時，方法四都比方法二快。

由表二和兩張圖可知，增加 K 可減少執行時間，尤其是在方法四時最明顯，而 K 是不能無限增加的，且在某些狀況下增加 K 後的執行時間不減反增，這在方法二和方法三中有部分有這種狀況，可能的原因為：K 會影響資料列的長度，K 越大每個資料段會越小，而在氣泡排序中因爲其時間複雜度大，所以資料列切得太長的話會耗費大量時間，但這也要考慮合併排序的時間複雜度，所以在每個方法和資料量下都會有一個 K 的甜蜜點。