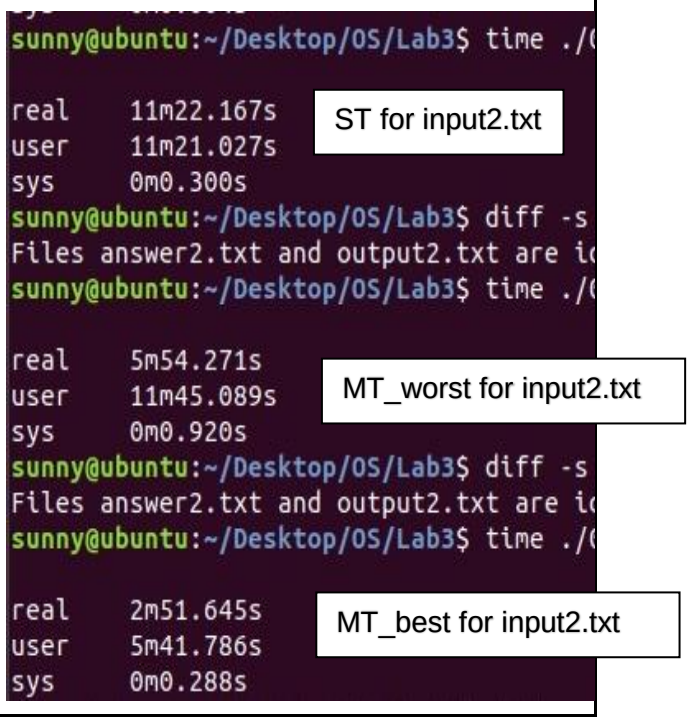
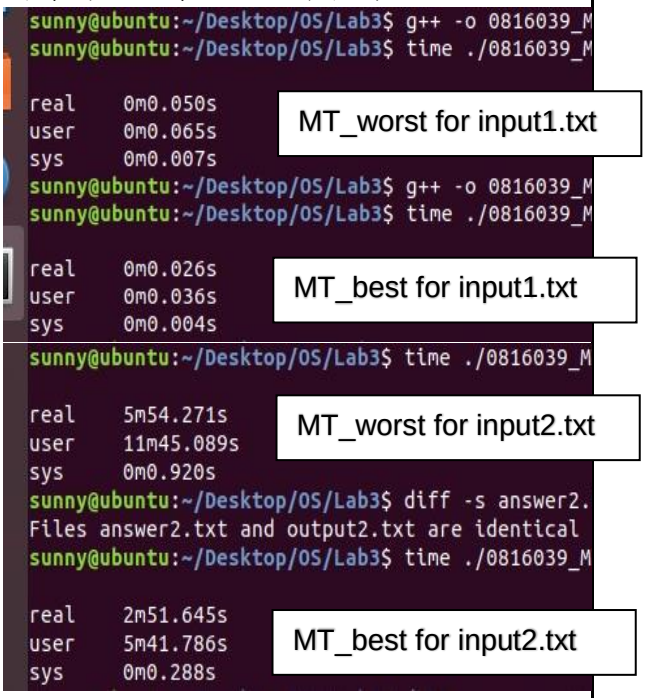
2021 NYCU OS HW3 report

Question	Answer
Q1. Briefly describe your design for the sorting algorithm, merge function, the thread management. Also, describing the number of sort threads and number of merge threads in the Multi-thread program.	首先在 sorting 的部分我用的是 insertion sorting，與一般 code 的地方不同的是，我們需要先把傳進來的 argument 分解成需要的 start 跟 end 再根據變數做 sorting。 Merge 的部分一樣會傳進 argument，我把它分成三類：start, mid, end，根據這三項變數依據決定要 merge 的兩個 vector 分別從哪裡開始。除了一般 merge 之外，我還在 multithread 的 best case 裡多定義了一個 merge_last function，避免最後要 merge 的部分取到舊的還沒做 sorting 的數字。 在 thread management 的部分我分別建立了 1, 3, 7 個 thread 的集合來處理每個 thread 的執行。 而我在 multithread program 中把 partition 分別設定為 2 等份（worst case, same as single thread）與 4 等份（best case），兩者我都只使用了兩個 thread。
Q2. Show the fastest time acceleration between single-thread and multi-thread. (Take screenshots of the time between single-thread and multi-thread)	

Q3. You need a brief description of the best multi-threads and worst multi-threads methods. The content includes the number of threads used and the way of partitioning, comparing the difference in time, and taking the screenshot between two multi-thread results.	如上面所述，我在 multithread 中皆用了 2 個 thread，worst case partition 成 2 等份，best case partition 成 4 等份。 將資料減為一半後，best case 相較於 worst case 執行時間減少了近一半。 不會變成 n^2 的原因是因為只有兩條 thread，所以 best case 的另外兩等份需要等前面兩等份做完才執行。  MT_worst for input1.txt MT_best for input1.txt MT_worst for input2.txt MT_best for input2.txt
Q4. What did you learn from doing hw3?	我學到要如何實際撰寫 thread 以及利用它能平行化的優點來加速程式的運行。 同時也回憶起怎麼從零開始寫 sorting、merging，和 structure。