B08501011\_PA1 report

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input size | IS | | MS | | QS | | HS | |
|  | CPU time(ms) | Memory(kB) | CPU time(ms) | Memory(kB) | CPU time(ms) | Memory(kB) | CPU time(ms) | Memory(kB) |
| 4000.case2 | 0.15 | 5904 | 1.22 | 5904 | 1.751 | 5904 | 0.8 | 5904 |
| 4000.case3 | 9.89 | 5904 | 1.265 | 5904 | 2.737 | 5904 | 1.093 | 5904 |
| 4000.case1 | 6.463 | 5904 | 1.666 | 5904 | 1.642 | 5904 | 1.424 | 5904 |
| 16000.case2 | 0.096 | 6056 | 2.012 | 6056 | 9.948 | 6056 | 1.883 | 6056 |
| 16000.case3 | 63.987 | 6056 | 2.171 | 6056 | 10.881 | 6056 | 1.611 | 6056 |
| 16000.case1 | 33.768 | 6056 | 3.784 | 6056 | 12.562 | 6056 | 3.785 | 6056 |
| 32000.case2 | 0.123 | 6188 | 3.95 | 6188 | 31.294 | 6188 | 3.065 | 6188 |
| 32000.case3 | 241.543 | 6188 | 2.467 | 6188 | 37.465 | 6188 | 2.146 | 6188 |
| 32000.case1 | 120.579 | 6188 | 5.935 | 6188 | 41.908 | 6188 | 4.185 | 6188 |
| 1000000.case2 | 1.098 | 12144 | 87.25 | 14004 | 80923 | 12144 | 75.89 | 12144 |
| 1000000.case3 | 252068 | 12144 | 85.647 | 14004 | 87555 | 12144 | 81.776 | 12144 |
| 1000000.case1 | 124247 | 12144 | 165.706 | 14004 | 88440 | 12144 | 148.7 | 12144 |

Run on EDA union workstation

1. Insertion sort:

最基本的排序方法，在已排序的狀況下有最快速度，且遠勝其他三種排序法，但在反向排序與亂序表現最差，表現起伏不穩定，且一般亂序情況速度也不好。

1. Merge Sort:

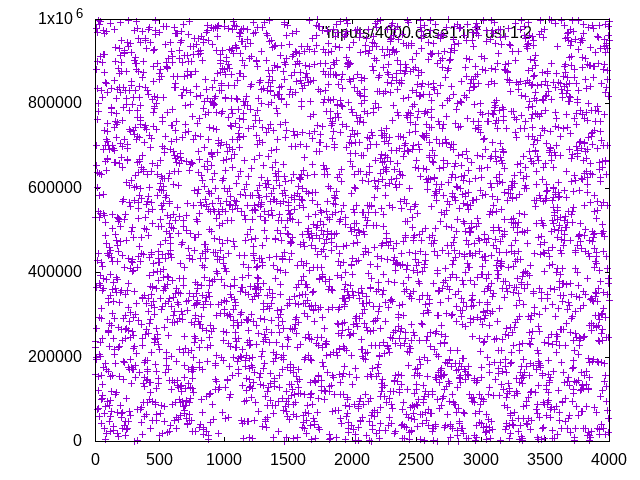
表現與Heap Sort程度相當，排序速度極快且對各種數據並沒有特別差異。另外推測由於電腦預開記憶體的原因，在小數據排序時並沒有辦法看出需要額外記憶體的特性，直到處理一百萬筆數據時才有記憶體使用較多的情形，

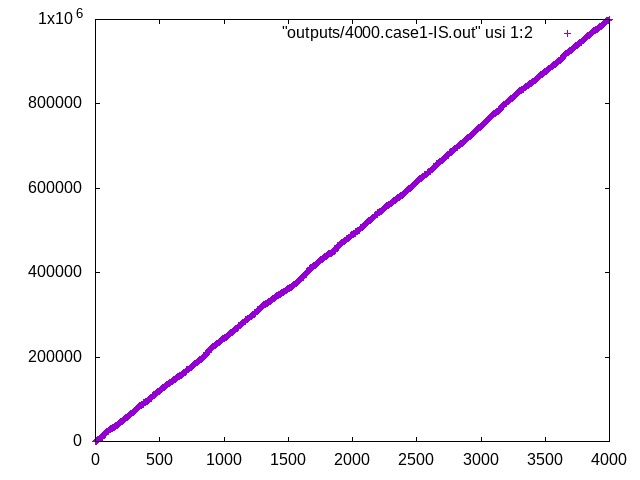
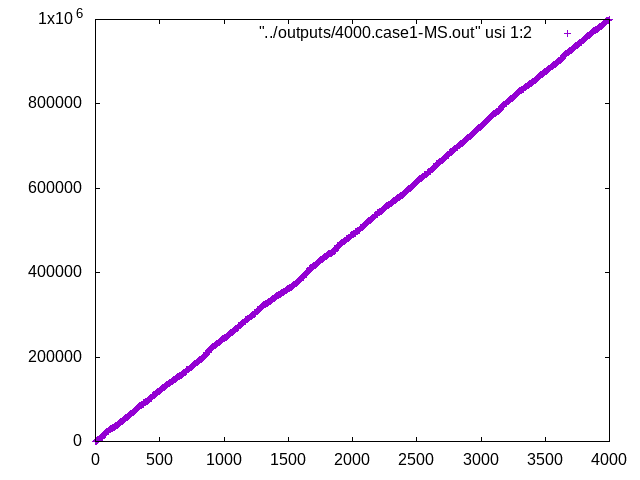
1. Quick Sort:

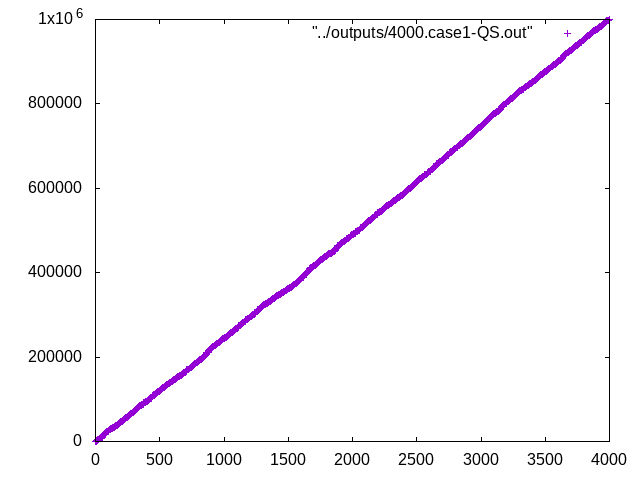
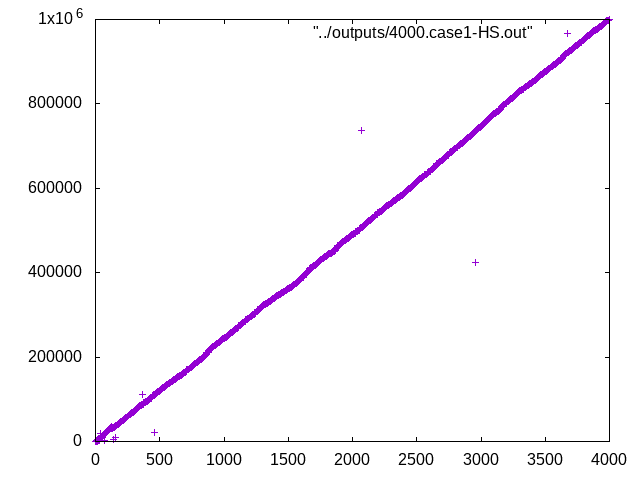
四種排序中處理較為不好的排序法，在寫程式時由於一直無法有效通過測試，故最後將pivot改為選取中間數而非最末端數，因此在分析Best case and worst case其實不太準確，造成上表所呈現的時間約略相等（皆為亂序），不過即使是亂序排序也並沒有很好的效率，推測應該是因為無法像Merge Sort一樣有效減少迭代次數，造成迭代次數過多而花費太多時間。

1. Heap Sort:

排序速度和Merge Sort相當且並不需要額外空間進行排序，表現最佳，需要注意的是在建立資料結構與寫程式時較為複雜。

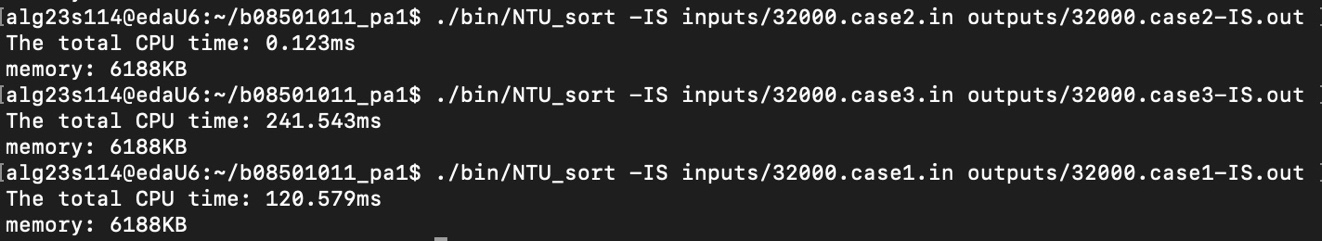
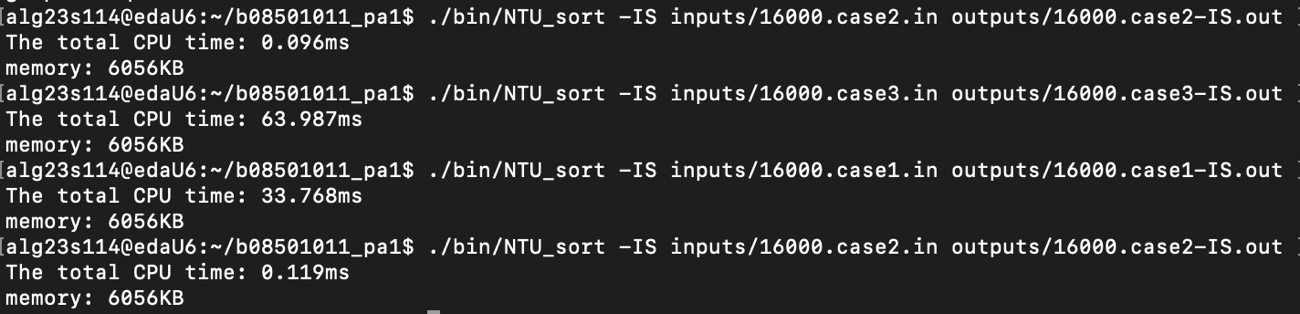
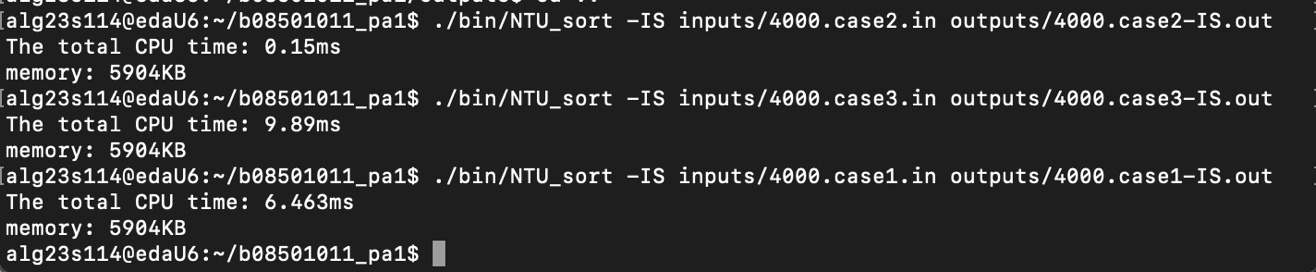


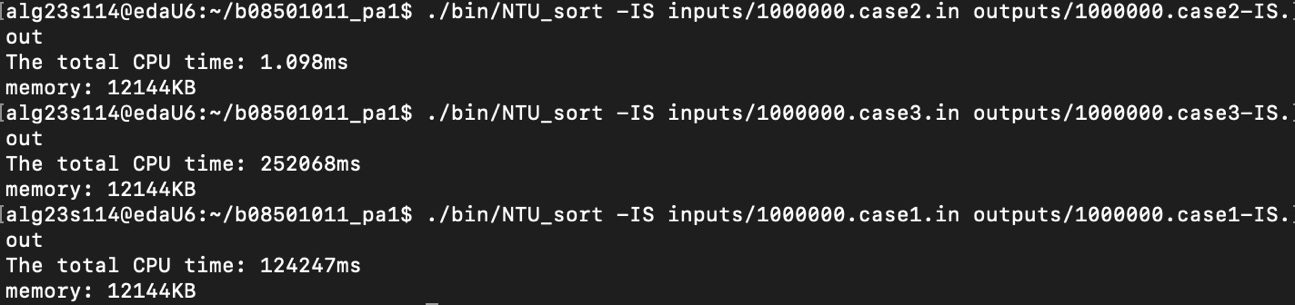
 

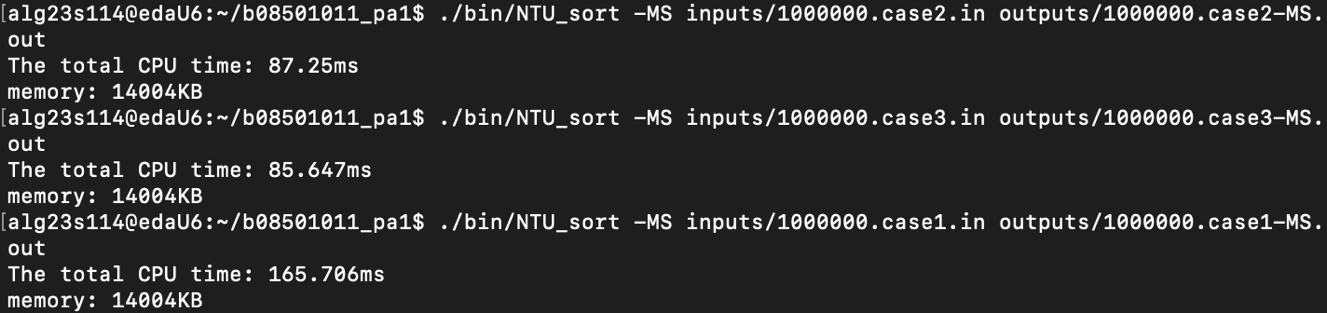
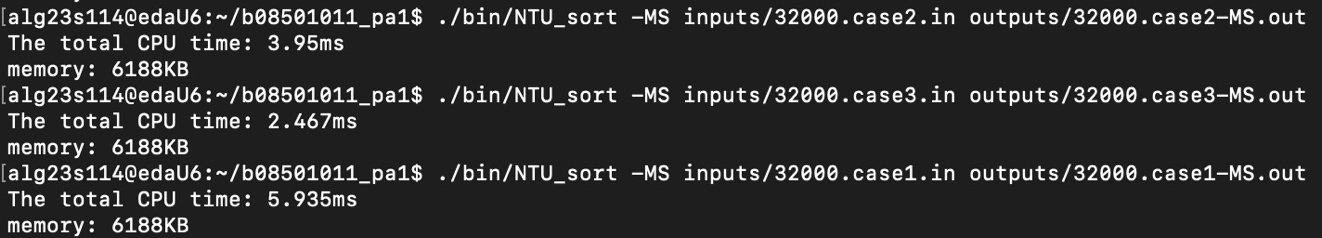
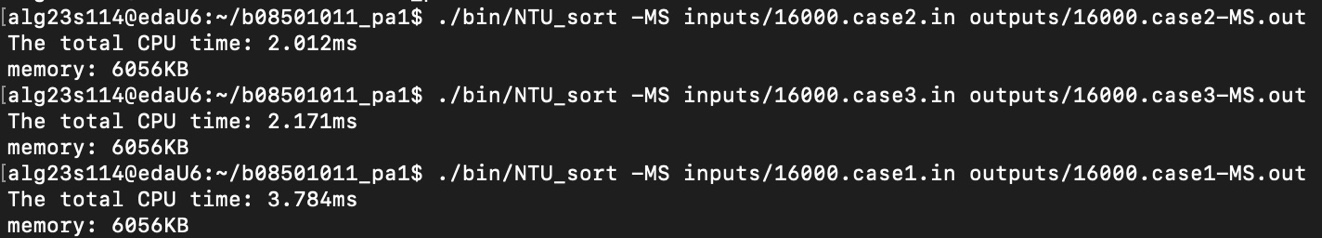
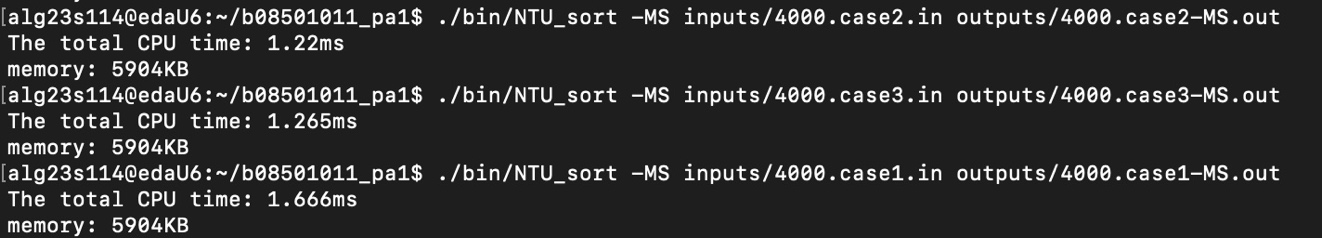
工作站執行結果截圖

1. Insertion Sort:

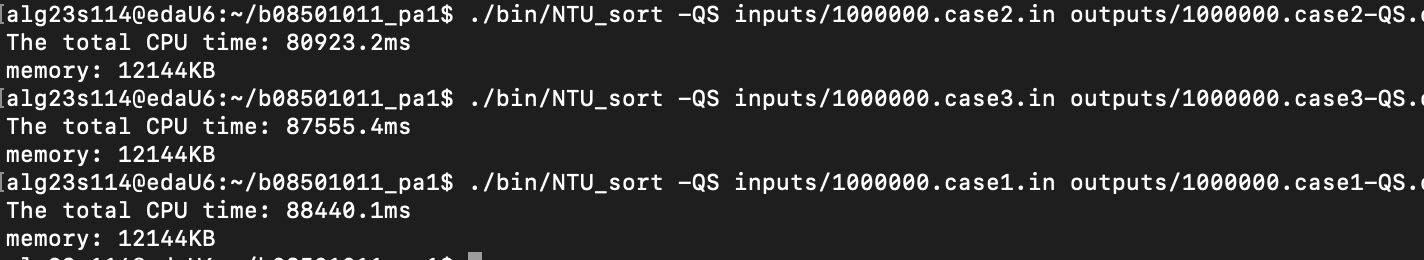
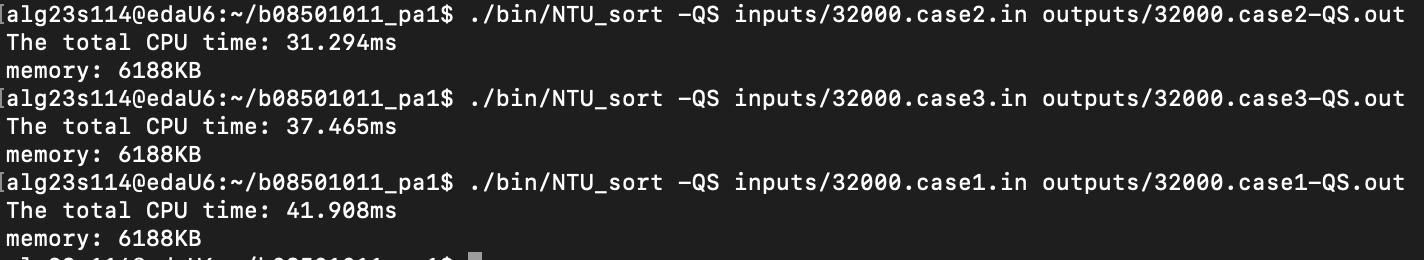
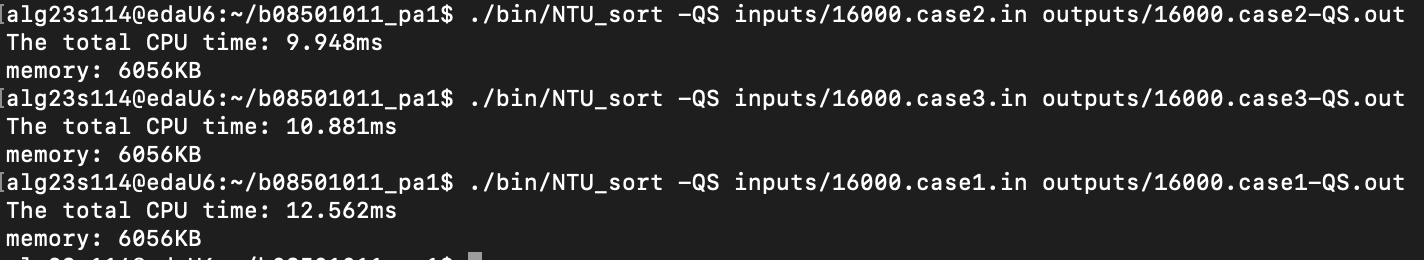
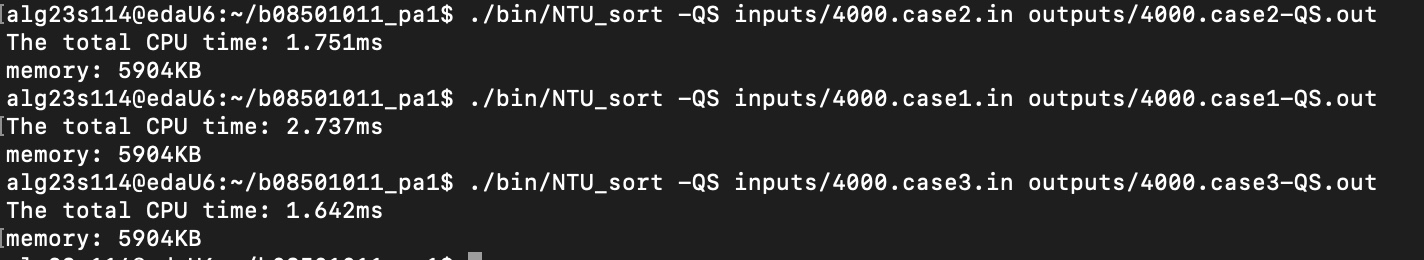




1. Merge Sort:



1. Quick Sort:



1. Heap Sort:

