# PA1 Report

徐子程
B06602037

## Test environment

CPU: Intel Xeon E3-1230 V2
RAM: DDR3 1866 32G
OS: Ubuntu 18.04.3 LTS
Compiler: g++ 7.4.0

## Test result

| 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 | 0.054 | 14016 | 0.897 | 14144 | 3.259 | 14016 | 0.293 | 14016 |
| 4000.case3 | 5.380 | 14016 | 0.561 | 14144 | 4.004 | 14016 | 0.293 | 14016 |
| 4000.case1 | 4.244 | 14016 | 0.606 | 14144 | 4.176 | 14016 | 0.374 | 14016 |
| 16000.case2 | 0.037 | 14152 | 2.905 | 14316 | 15.068 | 14160 | 0.924 | 14152 |
| 16000.case3 | 70.789 | 14152 | 2.580 | 14316 | 14.294 | 14160 | 0.892 | 14152 |
| 16000.case1 | 35.282 | 14152 | 3.815 | 14316 | 13.467 | 14160 | 1.793 | 14152 |
| 32000.case2 | 0.099 | 14284 | 3.974 | 14484 | 23.339 | 14292 | 2.048 | 14284 |
| 32000.case3 | 280.144 | 14284 | 3.316 | 14484 | 24.512 | 14292 | 2.002 | 14284 |
| 32000.case1 | 139.149 | 14284 | 5.061 | 14484 | 23.835 | 14292 | 3.338 | 14284 |
| 1000000.case2 | 0.888 | 20240 | 119.924 | 29924 | 723.380 | 20248 | 88.842 | 20240 |
| 1000000.case3 | 272675 | 20240 | 116.834 | 29924 | 696.966 | 20248 | 85.703 | 20240 |
| 1000000.case1 | 136253 | 20240 | 172.259 | 29924 | 735.552 | 20248 | 144.115 | 20240 |

※ The Quick Sort is implemented in “Randomized Quick Sort”.

## Test result analysis


Fig 1. Case 1 (random order)

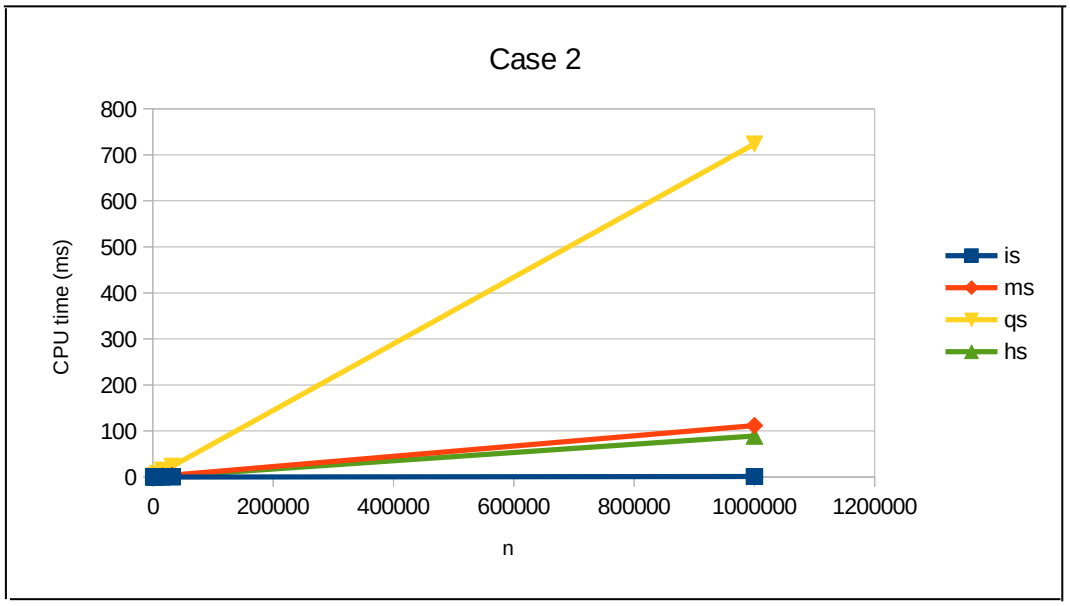


Fig 2. Case 2 (increasing order)

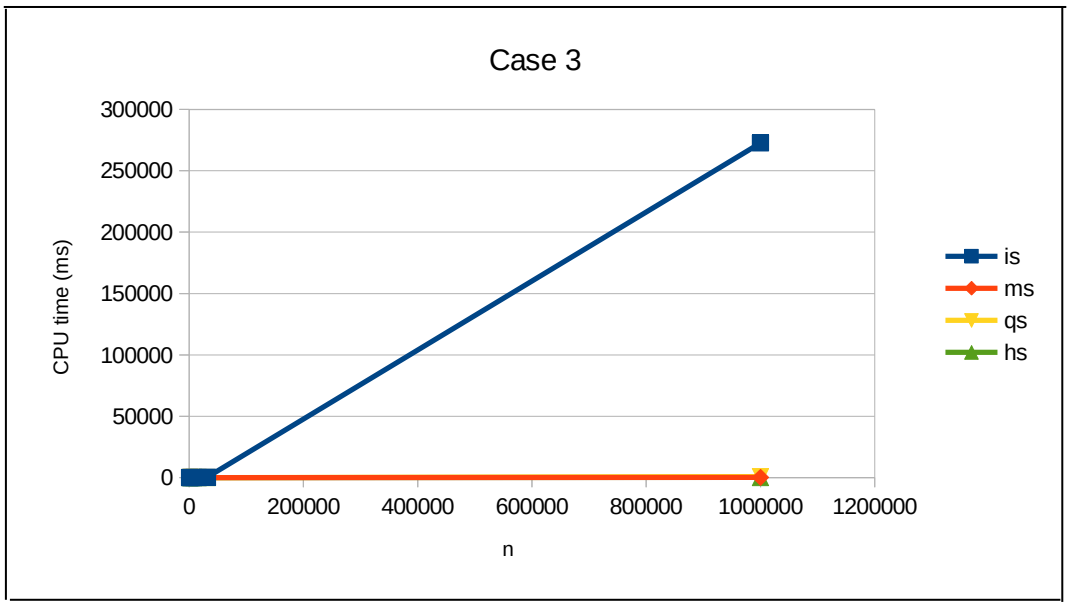


Fig 3. Case 3 (decreasing order)

For Insertion Sort, the order of input data has great effect on the running time. The running time in Case 3 is almost 2 times more than in Case 1 , where Case 3 is the worst case $O(n^2)$. However, in case 2, Insertion Sort is faster than other sorting algorithms (best case $O(n)$).

If we focus on other 3 sorting algorithms (QS, HS, MS), we will find that Merge Sort and Heap Sort has nearly the same performance ( $O(nlogn)$ ). Quick sort is slightly slower than the other 2 sorting algorithms. It may be a result of the random number generating process of Randomized Quick Sort, but this compromise effectively avoids the worst case $O(n^2)$.