tags: 資料結構

HW27

本篇作業需要比較四種不同 sort 在各自最壞複雜度的情況下所需的時間，首先會先介紹幾種排序方法各自的理論複雜度，最後在進行比較。

Analyze

Insertion sort

- 資料量少時較高效，且比其他的 $O(n^2)$ 的排序法還要高效
- 穩定排序，相同鍵值的元素排序後的相對位置不變
- 可在線處理，且不需要花費額外空間來排序

Status	Complexity
worst	$O(n^2)$
best	$O(n)$
average	$O(n^2)$
space	$O(1)$

Quick Sort

- 不穩定排序，相同鍵值的元素排序後的相對位置可能改變
- 需要花費額外空間來排序
- 分治算法

Status	Complexity
worst	$O(n^2)$
best	$O(n \log n)$
average	$O(n \log n)$
space	$O(\log n)$

需要注意的是 Quick sort 根據 pivot 的選擇不同會出現不同的複雜的，如果每次都選第一個元素當 pivot 的話在下列情況下就複雜度就會變得很糟糕。

1. 已經排序好的序列 (正序、倒序都一樣)
2. 全部元素都一樣 (第一種的特殊情況)

不過只需要將 pivot 改成中間位置的元素或隨機位置就能減少複雜度。

Merge Sort

- 最好、最壞、平均時間複雜度都是 $O(n \log n)$
- 穩定排序，相同鍵值的元素排序後的相對位置不變
- 分治算法

Status	Complexity
worst	$O(n \log n)$
best	$O(n \log n)$
average	$O(n \log n)$
space	$O(n)$

Heap Sort

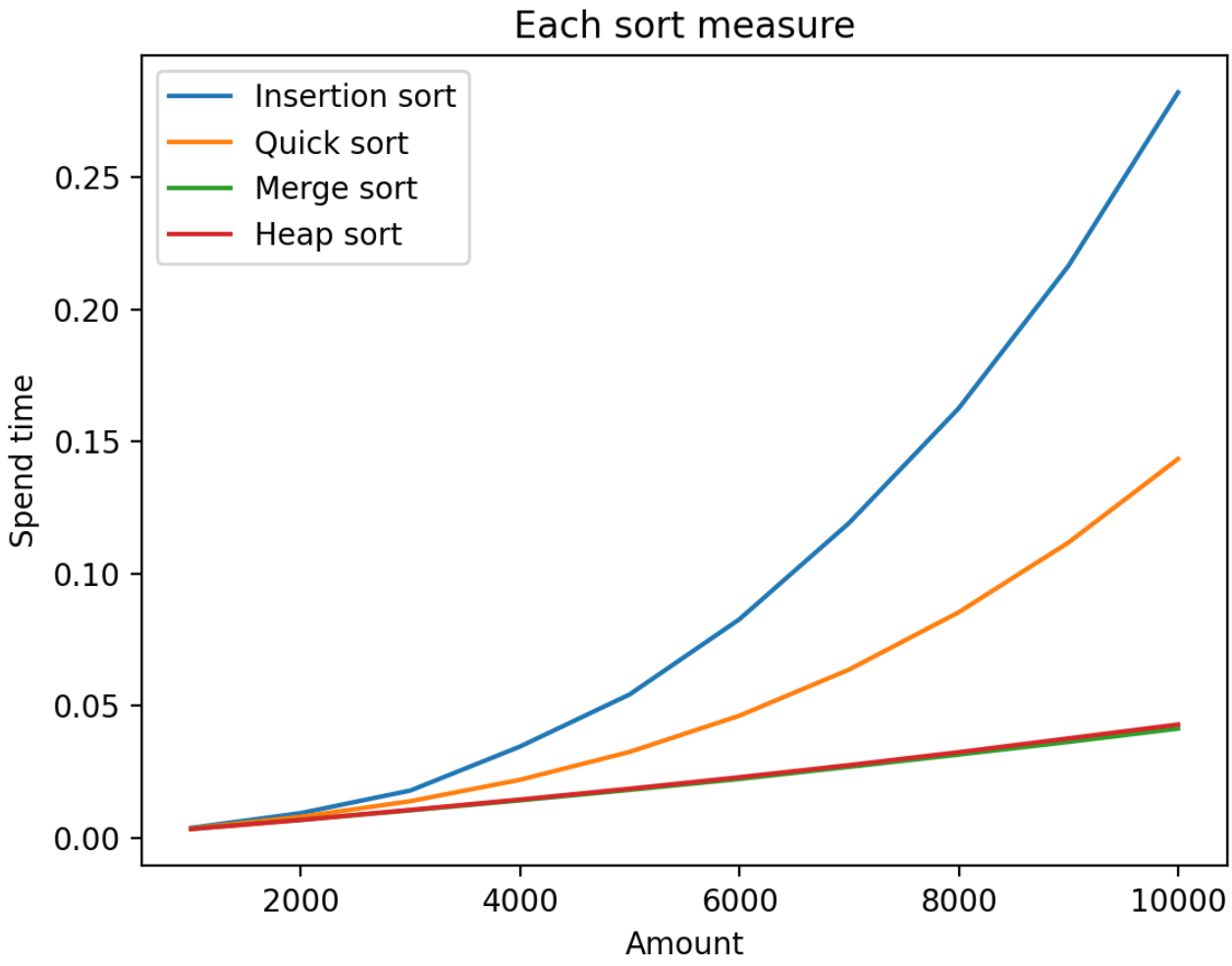
- 穩定排序，相同鍵值的元素排序後的相對位置不變
- 不需要花費額外空間來排序

Status	Complexity
worst	$O(n \log n)$
best	$O(n \log n)$
average	$O(n \log n)$
space	$O(1)$

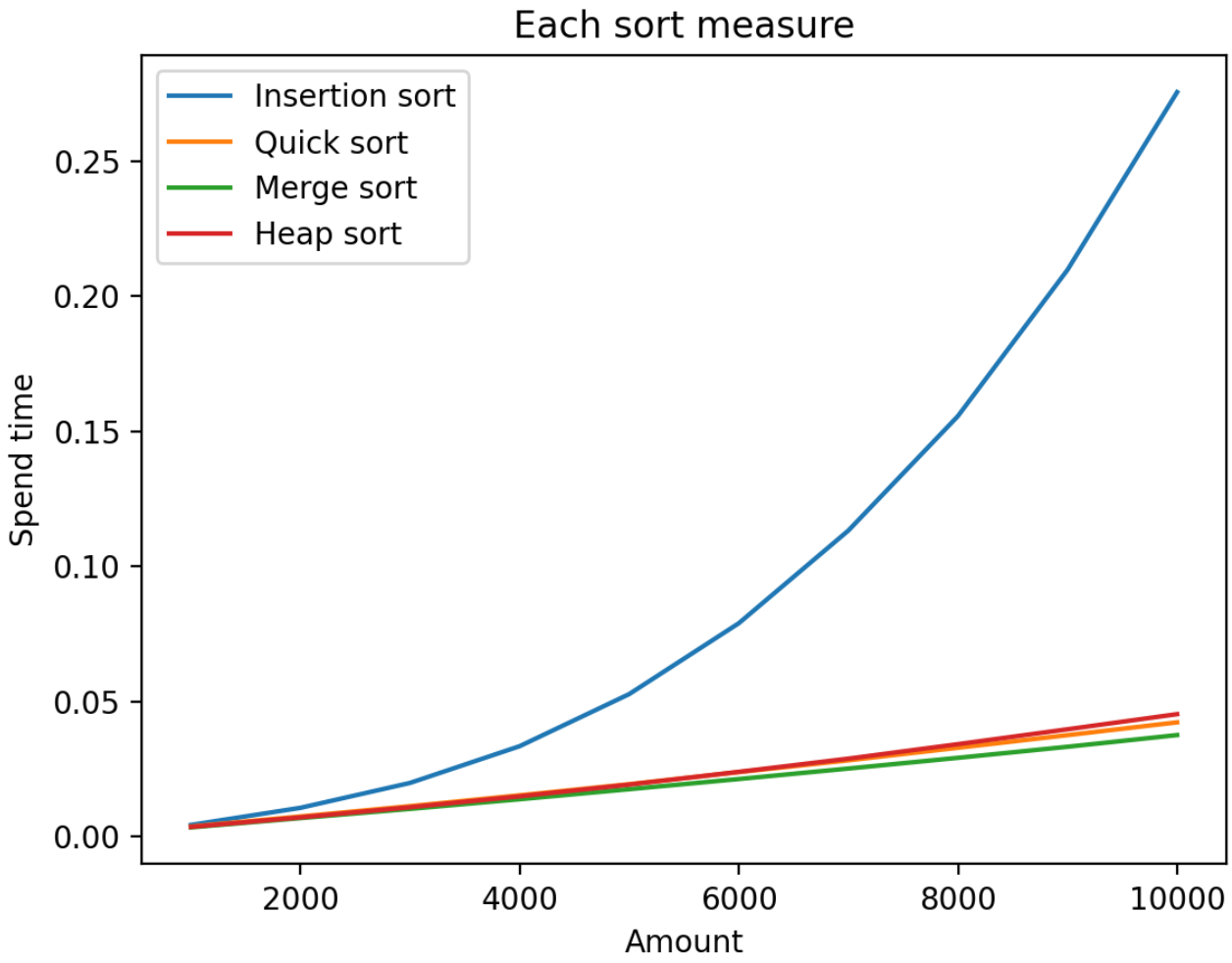
Measure

這邊衍伸出兩種比較，分別是 Quick sort pivot 選第一個，以及選擇中間那個，結果見下圖。

1. First element for pivot



2. Middle element for pivot



在測量結果中基本上可以看到最後結果大致符合預期的分佈，剩下僅有一下實作上的小誤差。