

Object Oriented Programming

P46091204 蔡承穎

HW05_03 report

下圖為 HW05_03.cpp 分別對三個.bin 檔與手寫演算法及利用 STL 中的 vector 的輸出結果，很明顯的，手寫演算法較快，以下將從結果做一個討論。

The result of HW5-3-1.bin

```
Time of Load Point : 17 ms  
Number of point 31632  
  
法一：手寫演算法測試(Quick sort)  
Cost Time 36 ms  
共有5276個點  
  
法二：用STL中Vector解  
Cost Time 56 ms  
共有5276個點  
  
法三：三層for迴圈解  
Cost Time 13814 ms  
共有5276個點
```

The result of HW5-3-2.bin

```
Time of Load Point : 1110 ms  
Number of point 4123500
```

```
法一：手寫演算法測試(Quick sort)  
Cost Time 6636 ms  
共有687166個點
```

```
法二：用STL中Vector解  
Cost Time 9597 ms  
共有687166個點
```

```
法三：三層for迴圈解
```

The result of HW5-3-3.bin

```
Time of Load Point : 5 ms  
Number of point 132
```

```
法一：手寫演算法測試(Quick sort)  
Cost Time 0 ms  
共有24個點
```

```
法二：用STL中Vector解  
Cost Time 0 ms  
共有24個點
```

```
法三：三層for迴圈解  
Cost Time 0 ms  
共有24個點
```

三層 For 迴圈時間複雜度是 $O(n^3)$ ，故是最慢的，以下不贅述這件事情。

STL 中 vector 的容器是可以動態增長的，但並不是每多加一個元素，就 reallocate 一次，因為記憶體的分配與釋放所耗資源是較大的，故要盡量減少 reallocate 的次數。然而 vector 作法是先初始化一個容量，當我們輸入的個數超過之，它就增長 1.5 或 2 倍。故當知道需要多少大小時，就指定 vector 大小較佳。

總而言之，個人總結手寫演算法較快的原因在於 vector 記憶體的分配與釋放所耗時間，因為 vector 增加/刪除元素的時間複雜度為 $O(n)$ ，排序為 $O(n\log n)$ ，利用 unique 函數的時間複雜度為 $O(n)$ ，相比較的，手寫演算法中快速排序法平均時間為 $O(n\log n)$ ，找重複點的 for loop 為 $O(n)$ ，兩者的時間複雜度差不多。