**作業二**

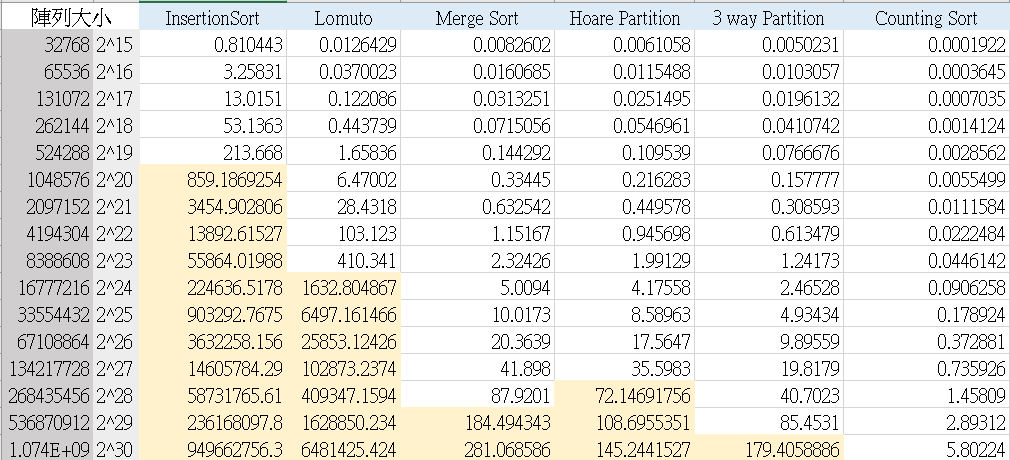
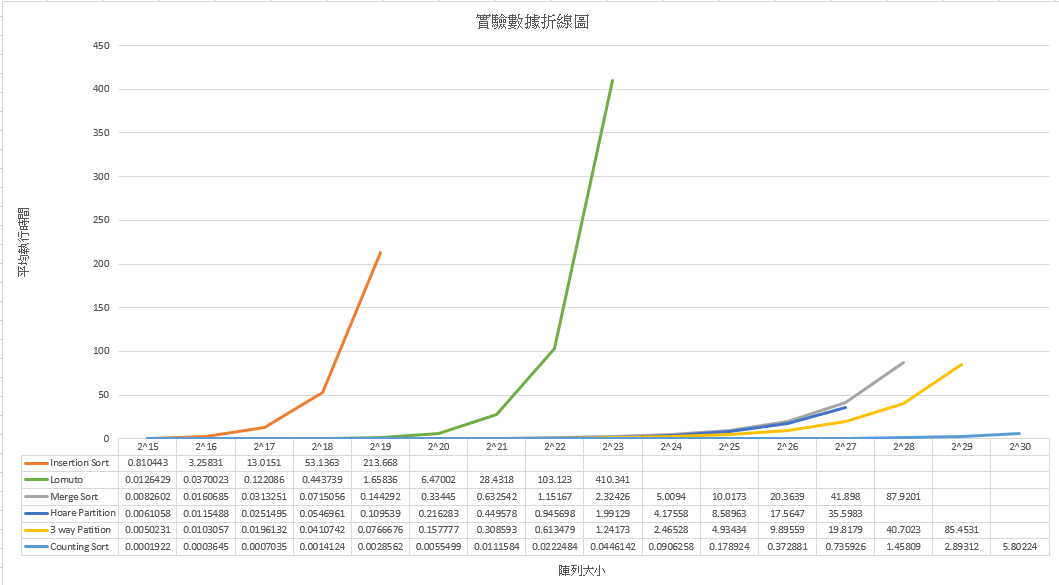
**109208064 資科三 俞懿**

**作業目標：**比較insertion sort, merge sort, 三種randomized quick sort (Lomuto Partition, Hoare Partition, 與3-Way Partition)與Counting sort。

**比較方法：**計算上述演算法在input array 大小為時（ = 15, 16, …, 30）的平均執行時間。針對每個input array 大小，分別產生10 個input array，最後取10 次執行時間的平均。若執行時間超過1小時或無法測試到，請估計執行時間，並解釋你的估計方法。

**Input array** 產生方法：陣列的每個數字都從1~1000 中的正整數中隨機選擇，且每個數字被選到的機率都相同（0.001）。

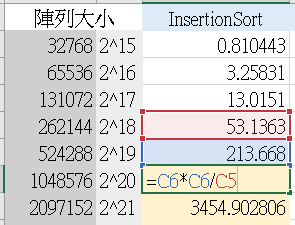
**繳交報告內容：**

1. 折線圖的x 軸為陣列大小，y 軸為對應的平均執行時間。表格中的數字為該陣列大小下，作10次input array insertion所花費的平均執行時間。
   * 實驗數據表格，黃色部分為預測數據
   * 由白色部分，真實實驗數據可以畫得以下折線圖

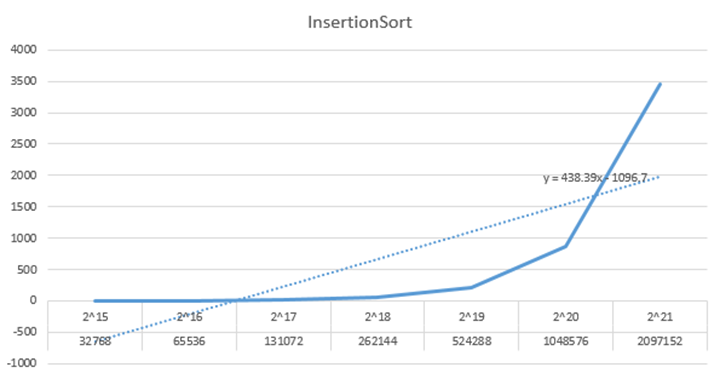
由排序array所花費的時間長短，可推得這些Sort的速度排名，時間花費越多代表這個sort越慢，時間花費越少代表這個sort越快。

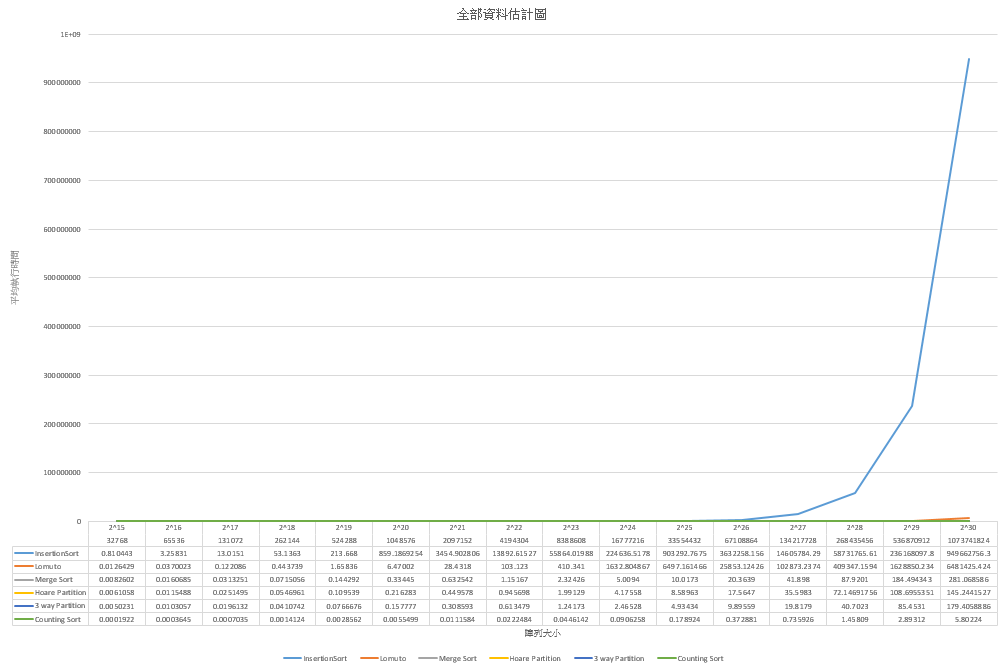
因此排名為 : Counting Sort -> 3 way Partition Quick Sort -> Hoare Partition Quick Sort-> Merge Sort -> Lomuto Partition Quick Sort -> Insertion Sort

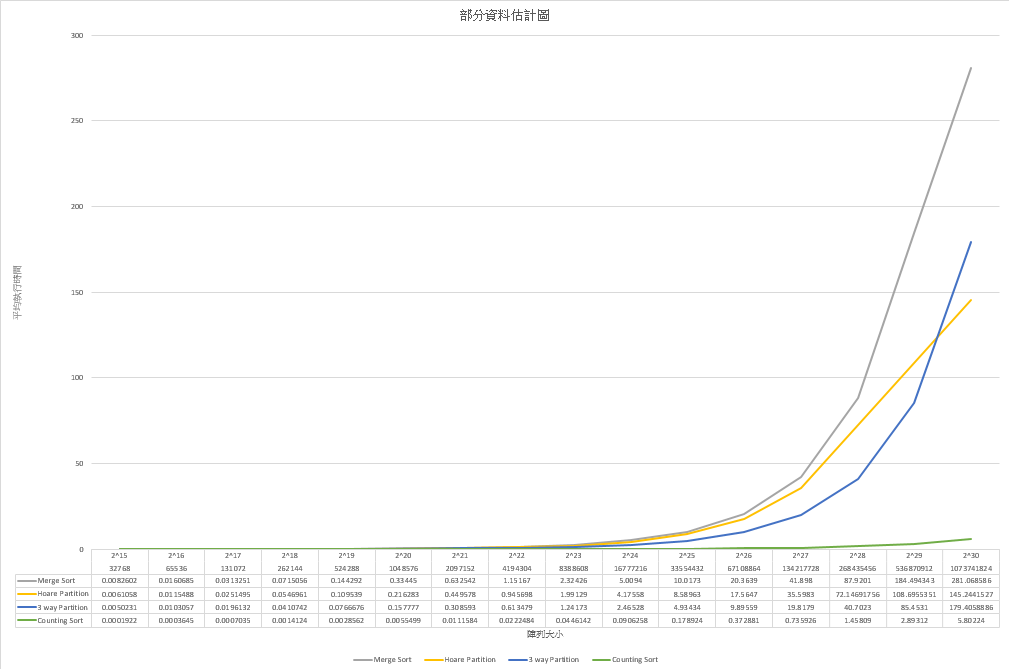
當執行時間超過1小時或無法測試到，數據無法跑出來，如上圖表格空白處，因此我以…方法估計執行時間。

* + 黃色部分的估計數值，我以倍數的方式計算。

因為若是以線性方程式估計，現有資料會視最後一筆平均執行時間為例外的特殊情況，線性方程式會無法表現出時間增長的快速，如下圖。



以前一筆資料乘以(前一筆資料的增加倍數)來做計算，得出的估計圖看起來比較合理一些，折線圖如下 : 



1. (10%) 每個Sorting 演算法的程式碼來源。
   * **Counting Sort :** <https://www.sanfoundry.com/cpp-program-implement-counting-sort>
   * **Insertion Sort :** <https://www.geeksforgeeks.org/insertion-sort>
   * **Merge Sort:** <https://www.geeksforgeeks.org/merge-sort/>
   * **3 way Partition Quick Sort :** <https://www.geeksforgeeks.org/3-way-quicksort-dutch-national-flag>
   * **Lomuto Partition Quick Sort :** <https://www.geeksforgeeks.org/hoares-vs-lomuto-partition-scheme-quicksort>
   * **Hoare Partition Quick Sort :** <https://www.geeksforgeeks.org/hoares-vs-lomuto-partition-scheme-quicksort>
2. (10%) 實驗的程式碼。

<https://github.com/yvonne90190/Data-Structure-HW/tree/main/hw2>

5. (20%) 心得、疑問、與遇到的困難

這個實驗要跑非常久，一開始覺得一個小時應該差不多可以跑完，沒想到好幾個sorting還是跑到k=17，接下來的數據就很慢，好像跑步出來依樣，因此決定讓電腦一直跑，在圖書館讀書的時候也跑，回宿舍睡覺之後還是讓他繼續跑，最後終於得到了一些比較接近k=30的數據，雖然Insertion Sort 和 Lomuto還離目標很遠，但這也證明了這幾個演算法很慢。

我覺得最困難的是預測數據，實在不知道該怎麼預測比較好，線性預測也不對、用excel畫指數的公式，代數字進去算卻一直error。上網看來一些統計學的估計方法，但也不是很懂這些估計方法的運作原理，最後只好以半猜的方式去畫圖找看看怎麼樣才合理。希望老師也可以指點一下我，該怎麼較準確地估計數據。