**Algorithms**

**Programming Assignment #1**

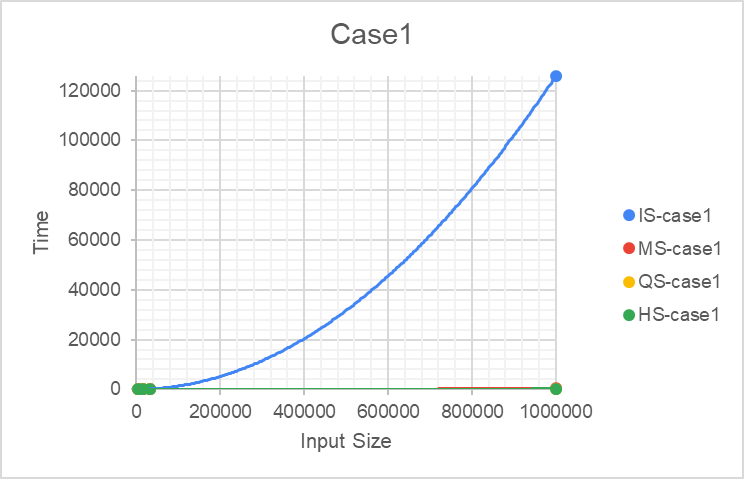
姓名：莊志淵 學號：r09921006

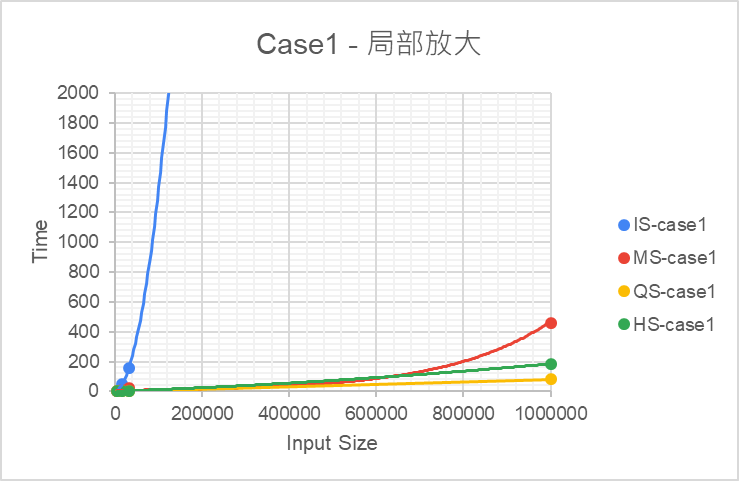
1. Data structures
   1. Insertion sort: 使用Array來implement List
   2. Merge sort: 使用Array來implement List
   3. Quick sort: 使用Array來implement List
   4. Heap sort: 使用Array來implement MaxHeap
2. Running time of four versions of different input sizes

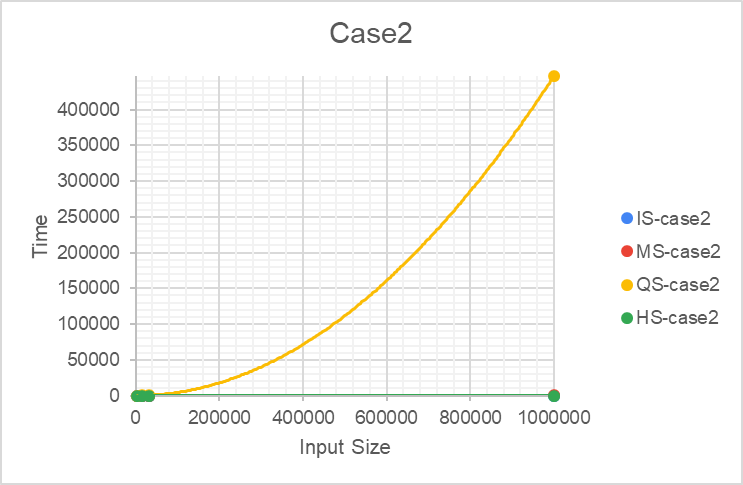


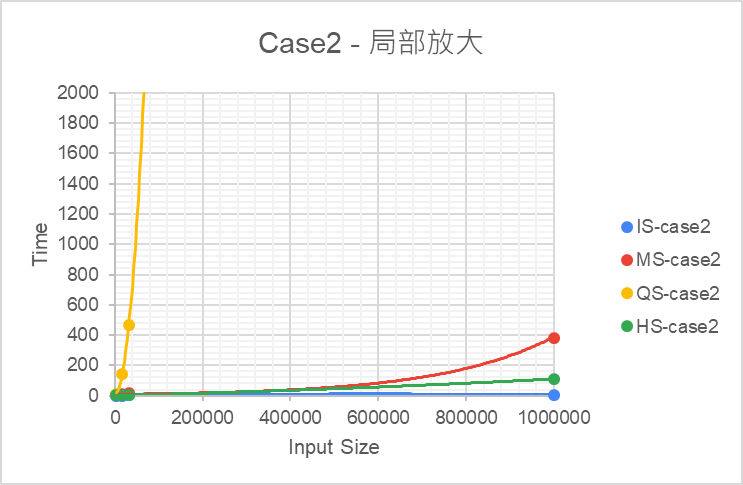
1. Figures to show the growth of running time as a function of input size

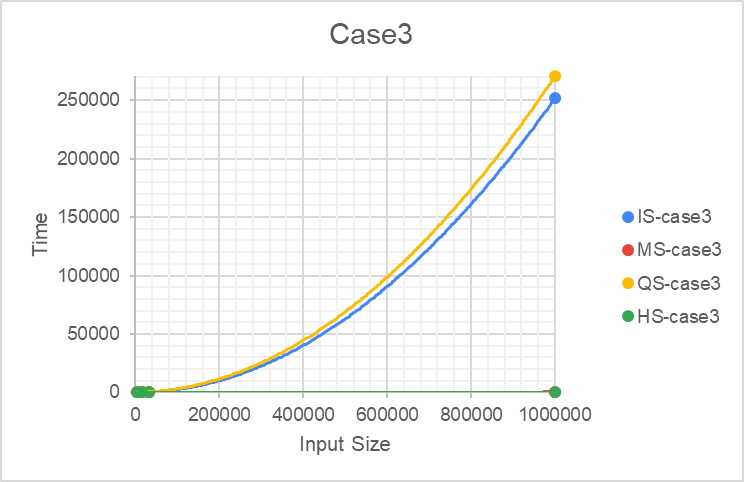
(Time unit is in ms) (For each case, 有原尺寸與局部放大的圖兩張)

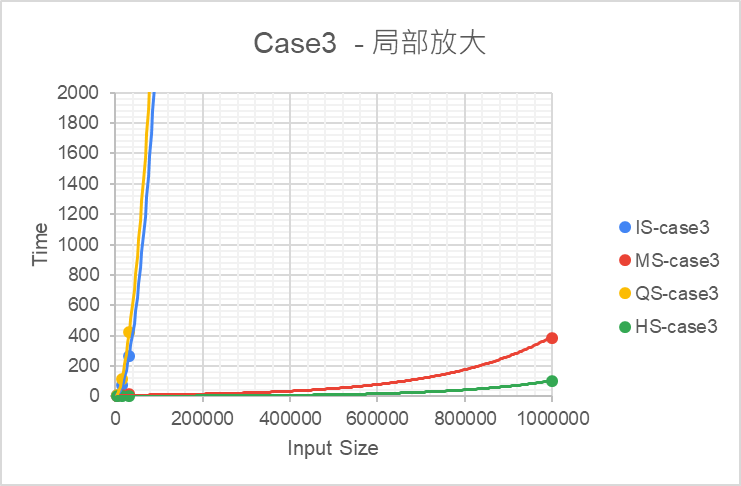












1. Analyze the curve
   1. Case1  
      case1是random order的input，基本上能夠視為average case，此時的time complexity可以由圖中看到，insertion sort的成長速度遠高於其他三者，由趨勢線可以驗證到insertion sort (IS)的time complexity為O(n2)，而其他三者(merge sort (MS), quick sort (QS), heap sort(HS)) 的time complexity為O(nlogn)，而在局部放大的圖中，也可以發現O(nlogn)的係數項大小應該是MS>HS>QS，不過這在random order的case1，應該是case by case的狀況。
   2. Case2  
      case2是increasing order的input，對於IS是best case，對於QS是worst case，因此也可以從圖中看出，IS的成長速度最慢，time complexity為O(n2)，而QS的成長速度最快，time complexity為O(n)，其餘MS與HS皆為O(nlogn)，其中MS的係數大於HS。
   3. Case3  
      case3是reverse order的input，對於IS與QS都是worst case，因此可以從圖中看出，IS與QS的time complexity都是O(n2)，而MS與HS是O(nlogn)，其中MS的係數大於HS。
2. Findings in this programming assignment

在課堂中學到各個sorting方式的asymptotic time complexity，透過實際的操作，能發現即使在big-O上具有相同order，但是對實際精確的time complexity仍有所差距。同時space complexity也是需要考量的，worst case下，透過表格可見得，QS是O(nlogn)最差，MS約是O(n)其次。