

I. Run Time

1. Size 為 100 的數據

```
Microsoft Visual Studio Debu  X + -
Enter the size of the array: 100
Generated array(by insertion sort):
87 223 302 437 895 1076 1872 2385 2711 3360 3413 4000 4106 4682 6007 6079 6284 6310 6674 6696 7430 7596 8774 8863 9478 9
586 9950 9981 10915 10937 11313 11844 13121 13344 13468 13781 13830 14034 14066 14274 14352 14627 14721 15620 15894 1601
1 16171 16376 16907 17218 17223 17337 17394 17665 17884 18788 19039 19402 20324 20432 20435 20551 20689 20926 21647 2165
2 21813 21900 22038 22086 22365 22691 22885 23010 23534 23575 24190 24264 24828 25035 25286 25834 25958 26339 26414 2659
8 26628 26773 26795 29257 29923 30066 30272 30639 30687 30791 31708 32135 32263 32384
insertion-sorting run time:0
Generated array(by merge sort):
87 223 302 437 895 1076 1872 2385 2711 3360 3413 4000 4106 4682 6007 6079 6284 6310 6674 6696 7430 7596 8774 8863 9478 9
586 9950 9981 10915 10937 11313 11844 13121 13344 13468 13781 13830 14034 14066 14274 14352 14627 14721 15620 15894 1601
1 16171 16376 16907 17218 17223 17337 17394 17665 17884 18788 19039 19402 20324 20432 20435 20551 20689 20926 21647 2165
2 21813 21900 22038 22086 22365 22691 22885 23010 23534 23575 24190 24264 24828 25035 25286 25834 25958 26339 26414 2659
8 26628 26773 26795 29257 29923 30066 30272 30639 30687 30791 31708 32135 32263 32384
merge-sorting run time:1
heapsorting run time:0
87 223 302 437 895 1076 1872 2385 2711 3360 3413 4000 4106 4682 6007 6079 6284 6310 6674 6696 7430 7596 8774 8863 9478 9
586 9950 9981 10915 10937 11313 11844 13121 13344 13468 13781 13830 14034 14066 14274 14352 14627 14721 15620 15894 1601
1 16171 16376 16907 17218 17223 17337 17394 17665 17884 18788 19039 19402 20324 20432 20435 20551 20689 20926 21647 2165
2 21813 21900 22038 22086 22365 22691 22885 23010 23534 23575 24190 24264 24828 25035 25286 25834 25958 26339 26414 2659
8 26628 26773 26795 29257 29923 30066 30272 30639 30687 30791 31708 32135 32263 32384

C:\Users\40924\Desktop\C\Algorithm_HW03\Debug\Algorithm_HW03.exe (process 15488) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

2. Size 為 1000 的數據

```
Microsoft Visual Studio Debu  X + -
Enter the size of the array: 1000
insertion-sorting run time:10
merge-sorting run time:13
heapsorting run time:1

C:\Users\40924\Desktop\C\Algorithm_HW03\Debug\Algorithm_HW03.exe (process 3984) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

3. Size 為 10000 的數據

```
Microsoft Visual Studio Debu  X + -
Enter the size of the array: 10000
insertion-sorting run time:1006
merge-sorting run time:129
heapsorting run time:23

C:\Users\40924\Desktop\C\Algorithm_HW03\Debug\Algorithm_HW03.exe (process 24676) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

4. Size 為 100000 的數據

```
Microsoft Visual Studio Debug Console
Enter the size of the array: 100000
insertion-sorting run time:96636
merge-sorting run time:1425
heapsorting run time:288

C:\Users\40924\Desktop\C\Algorithm_HW03\Debug\Algorithm_HW03.exe (process 16684) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

5. Size 為 1000000 的數據(因為 insertion sort 的 runtime 會很大，所以省略)

```
Microsoft Visual Studio Debug Console
Enter the size of the array: 1000000
merge-sorting run time:13661
heapsorting run time:3648

C:\Users\40924\Desktop\C\Algorithm_HW03\Debug\Algorithm_HW03.exe (process 12340) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

II. 分析：

- 對於較小的 Size，演算法 Insertion Sort 的 runtime 較 Merge Sort 快；但隨著數字越來越大，兩者的差距將會拉近，最後 Merge Sort 的速度會大幅超前 Insertion Sort；而 Heapsort 的 runtime 始終很小，隨著數字越來越大，Heapsort 在時間上的優勢越來越明顯。
- 結論並不難想像，Merge Sort 要先經過 Merge 的過程，但對於 array size 很小的數列，先將數列分到剩一個元素這個動作反而有些多餘，不如 Insertion Sort 一個一個比對踏實；相比之下，Size 變大時，Insertion Sort 就失去了優勢；Heapsort 在 size 較小卻仍優於其他兩者的原因可能有以下幾種可能：
 - Insertion sort 在 best case 的情況下時間複雜度為 $O(n)$ ，但我們沒有辦法保證每次 random 產生的數列都有部分排序的狀況；Heapsort 雖然在 build heap 的過程需要一些時間，但可能是因為有更好的存取方式，而執行的更快。
 - Heapsort 的一個 loop 只會有父子，但 insertion sort 一次 loop 就幾乎包含整個數列，就算數列有些部分已經排序完成。
 - Heapsort 在記憶體存取模式可能更加高效。
- 換句話說，Insertion Sort 比較節省運算的空間；Merge Sort 對於運算的時間比較有效率；Heap sort 節省了運算的空間，運算時間複雜度也跟 merge sort 一樣，但它需要額外內存空間。

III. 時間複雜度：

1. 理論：

In Insertion Sort, $O(f(n)) = O(n^2)$.

In Merge Sort, $O(f(n)) = O(n \log n)$.

In Heapsort, $O(f(n)) = O(n \log n)$.

2. 實際：

Size 每增加 10 倍，Insertion Sort 的 run time 增加 100 倍；Size 每增加 10 倍，Merge Sort 的 run time 增加 10 倍。已知 $10^2 = 100$ ， $10 \log 10 = 10 \times 1 = 10$ ，隨著輸入大小 n 的增加，運行時間按照 $n \log n$ 的速率增長，由此可知此程式的複雜度符合 $O(f(n))$ 。Size 每增加 10 倍，Heapsort 的 run time 增加 12 倍左右，可能是因為隨著 array size 變大，compare 和 swap 操作比較多，所以 runtime 會較大。

IV. 心得

上次的作業因為有太多筆數據，程式跑 data.txt 會跑十五分鐘都跑不出來。面對多個 maximum subarray 的序列時，我也沒有想到很好的方法去讓運算時間快一點，反而加上考慮多個 maximum subarray 的程式之後，運行時間拉得更久。這次作業相對簡單了很多，了解原理後很快就寫完了，我反而花了比較多時間分析這三種 sorting 的差異，希望下次遇到比較困難的 homework 的時候可以多融會貫通一點。