

I. Run Time

1. Size 為 100 的數據

```
Microsoft Visual Studio Debug Console
Enter the size of the array: 100
Generated array(by insertion sort):
247 534 1013 1194 1234 1508 1665 2959 3212 3281 4563 5278 5602 5799 5866 5948 6203 6619 6634 6710 7046 7502 7878 8369 84
31 9276 9392 9591 10072 10997 11032 11667 11903 11942 12595 13019 13764 13765 13931 14287 14388 15030 15268 15464 15612
15648 16717 16762 17766 18190 18377 18696 18928 19125 19449 19676 19696 19817 20476 20557 21145 21397 21742 21843 22437
22459 22619 22731 23125 23378 23591 23784 24031 24087 24476 24587 24930 25436 26042 27265 27611 27802 28952 29335 29634
29786 29849 30253 30880 30913 31278 31287 31532 32191 32408 32427 32516 32565 32620 32624
insertion-sorting run time:0
Generated array(by merge sort):
247 534 1013 1194 1234 1508 1665 2959 3212 3281 4563 5278 5602 5799 5866 5948 6203 6619 6634 6710 7046 7502 7878 8369 84
31 9276 9392 9591 10072 10997 11032 11667 11903 11942 12595 13019 13764 13765 13931 14287 14388 15030 15268 15464 15612
15648 16717 16762 17766 18190 18377 18696 18928 19125 19449 19676 19696 19817 20476 20557 21145 21397 21742 21843 22437
22459 22619 22731 23125 23378 23591 23784 24031 24087 24476 24587 24930 25436 26042 27265 27611 27802 28952 29335 29634
29786 29849 30253 30880 30913 31278 31287 31532 32191 32408 32427 32516 32565 32620 32624
merge-sorting run time:2
Generated array(by heapsort):
247 534 1013 1194 1234 1508 1665 2959 3212 3281 4563 5278 5602 5799 5866 5948 6203 6619 6634 6710 7046 7502 7878 8369 84
31 9276 9392 9591 10072 10997 11032 11667 11903 11942 12595 13019 13764 13765 13931 14287 14388 15030 15268 15464 15612
15648 16717 16762 17766 18190 18377 18696 18928 19125 19449 19676 19696 19817 20476 20557 21145 21397 21742 21843 22437
22459 22619 22731 23125 23378 23591 23784 24031 24087 24476 24587 24930 25436 26042 27265 27611 27802 28952 29335 29634
29786 29849 30253 30880 30913 31278 31287 31532 32191 32408 32427 32516 32565 32620 32624
heapsorting run time:0
Generated array(by randomized quicksort):
247 534 1013 1194 1234 1508 1665 2959 3212 3281 4563 5278 5602 5799 5866 5948 6203 6619 6634 6710 7046 7502 7878 8369 84
31 9276 9392 9591 10072 10997 11032 11667 11903 11942 12595 13019 13764 13765 13931 14287 14388 15030 15268 15464 15612
15648 16717 16762 17766 18190 18377 18696 18928 19125 19449 19676 19696 19817 20476 20557 21145 21397 21742 21843 22437
22459 22619 22731 23125 23378 23591 23784 24031 24087 24476 24587 24930 25436 26042 27265 27611 27802 28952 29335 29634
29786 29849 30253 30880 30913 31278 31287 31532 32191 32408 32427 32516 32565 32620 32624
randomized quicksort run time:0

C:\Users\40924\Desktop\C\Alg_HW4\Debug\Alg_HW4.exe (process 23556) 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 . . .
```

2. Size 為 1000 的數據

```
Microsoft Visual Studio Debug Console
Enter the size of the array: 1000
insertion-sorting run time:12
merge-sorting run time:14
heapsorting run time:1
randomized quicksort run time:1

C:\Users\40924\Desktop\C\Alg_HW4\Debug\Alg_HW4.exe (process 17360) 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 . . .
```

3. Size 為 10000 的數據

```
Microsoft Visual Studio Debu x + v
Enter the size of the array: 10000
insertion-sorting run time:1008
merge-sorting run time:133
heapsorting run time:25
randomized quicksort run time:15

C:\Users\40924\Desktop\C\Alg_HW4\Debug\Alg_HW4.exe (process 24324) 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 . . .
```

4. Size 為 100000 的數據

```
Microsoft Visual Studio Debu x + v
Enter the size of the array: 100000
insertion-sorting run time:99854
merge-sorting run time:1331
heapsorting run time:285
randomized quicksort run time:182

C:\Users\40924\Desktop\C\Alg_HW4\Debug\Alg_HW4.exe (process 17240) 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：

Randomized Quicksort 和 Merge sort 兩種排序演算法效率都很高，Randomized Quicksort 的優點是執行較為簡單，通常具有比較好的性能，另外也不需要額外的記憶體儲存空間；Merge Sort 的穩定性較好，時間複雜度也不會受原數列排序的分布影響。

由此可知，如果對排序的穩定性有較高的需求，或無法原地重新排列陣列，Merge Sort 會比較適用；對於需要實現較簡單或是性能要求較高的狀況，可以用 Randomized Quicksort，只是要注意可能會有 worst case。

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

In Randomized Quicksort, $O(f(n)) = O(n^2)$ in worst case;
 $O(f(n)) = O(n \log n)$ in average case.

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。

相比之下，Randomized Quicksort 的 runtime 就沒有規律，這是因為在平均情況下，快速排序法將數列大致分為相等的兩部份時， $O(f(n)) = O(n \log n)$ 才會成立，但大多數情況下，隨機抽出數列中的元素並不會剛好將數列切一半，甚至會有 worst case 產生。

IV. 心得

這次的演算法我覺得比上次的作業還容易理解，partition 副程式的 exchange 剛開始搞錯順序，花了一點時間 debug，其他沒什麼問題，randomized quicksort 跟一般的 quicksort 差不多，只是多了一個隨機選數的過程。