

Programming Assignment #1:

Indexing and Sorting Report

B05901182 電機三 潘彥銘

一、演算法流程(Algorithm Flow)

- 1. Insertion sort:** 從未經過排列的數列中取出一個元素，由後往前與已經排列好數列一一比較，直到遇到不大於自己的元素，並插入在此元素之後。重複上述的步驟直到整個數列都排序好。
- 2. Merge sort:** 將一個未排序的數列分割成若干個子數列，分割到子數列的大小符合當初所設的初始條件(起始數字等於結尾數字)後，將子數列與另一子數列合併並排序，重複合併到整個數列都排序好。
- 3. Heap sort:** 將整個數列變成max-heap的形式後，把數列的第一個與最後一個交換，把整個數列扣除最後一個並且再將此數列變成max-heap的形式，重複此步驟直到整個數列都排序好。
- 4. Quick sort:** 拿數列的第一個當作標兵，比標兵小的放標兵左邊，比標兵大的放標兵右邊，之後將標兵左邊與右邊分別視為一個新的數列，並再各選一個標兵，重複上述步驟直到整個數列都排序好。

二、資料結構 (Data Structure)

```
24     typedef vector< pair<string, int> > VP;  
25     VP vp;  
26     vp.reserve(p.QueryTotalStringCount());
```

圖(一)

1. 此次作業的四個sort我都是用圖(一)第24-25行的資料結構，先以pair儲存字串(string)與字串出現的字序(int)。
2. 由於vector算是一種動態陣列，為了加速，我在第26行先設定好vector的大小。

```
28     for ( int i = 0 ; i < p.QueryTotalStringCount() ; i++ ) {  
29         vp.push_back(make_pair(p.QueryString(i), i+1)); }
```

圖(二)

3. 將已經被parser處理好的字串與字串出現的字序一一做成pair，並存入在圖(一)定義好的vector裡。

三、問題與討論 (Discussion)

(1)實作中遇到的問題

```
45 int partition(VP& ex, int p, int r) {  
46     srand( time(NULL) );  
47     pair<string, int> x = ex[(rand()%(r-p+1))+p];
```

圖(三)

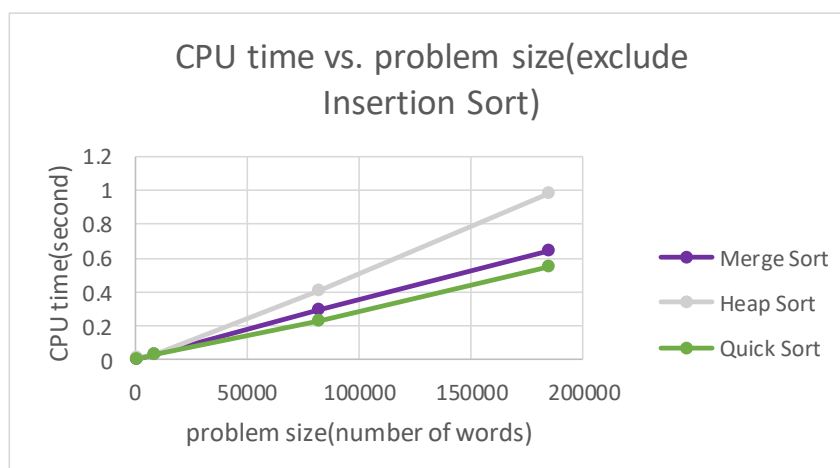
1. 在實作**quick sort**時，我嘗試不用課程講義上以數列第一個做為標兵，我試著用隨機的方式選取標兵，圖(三)是我部分code的截圖，**p**是數列的第一個index，**r**是數列的最後一個index，然而在此種選標兵的方式下，同樣的測資，有時候可以跑出結果，有時候卻不行，問題原因仍在討論中。

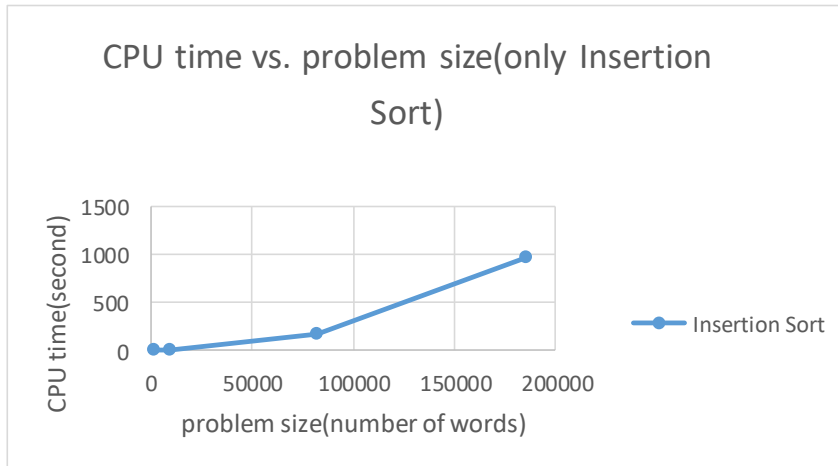
(2)討論

1. CPU time and memory usage table

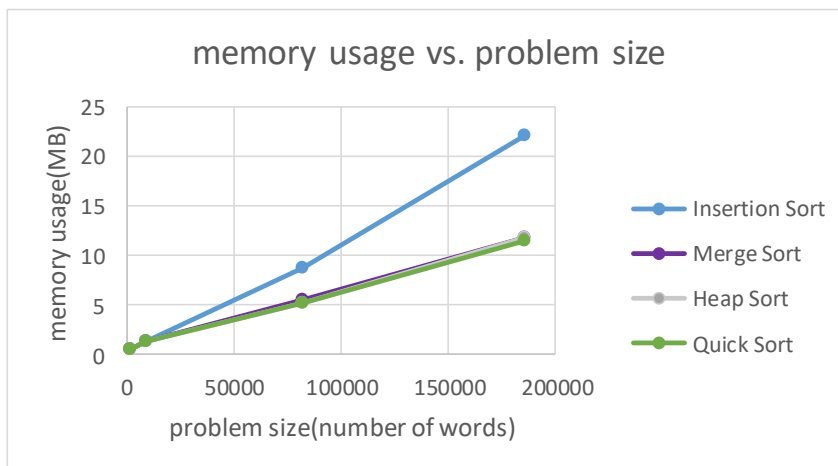
Test Case	Insertion Sort		Merge Sort		Heap Sort		Quick Sort	
	Time (sec)	Memory (MB)	Time (sec)	Memory (MB)	Time (sec)	Memory (MB)	Time (sec)	Memory (MB)
Case1	0.093	0.6	0	0.6	0.015	0.6	0	0.6
Case2	2.171	1.4	0.031	1.3	0.031	1.3	0.031	1.3
Case3	176.25	8.7	0.297	5.5	0.406	5.3	0.234	5.3
Case4	966.17	22.2	0.64	11.8	0.985	11.8	0.547	11.5

2. CPU time vs. problem size



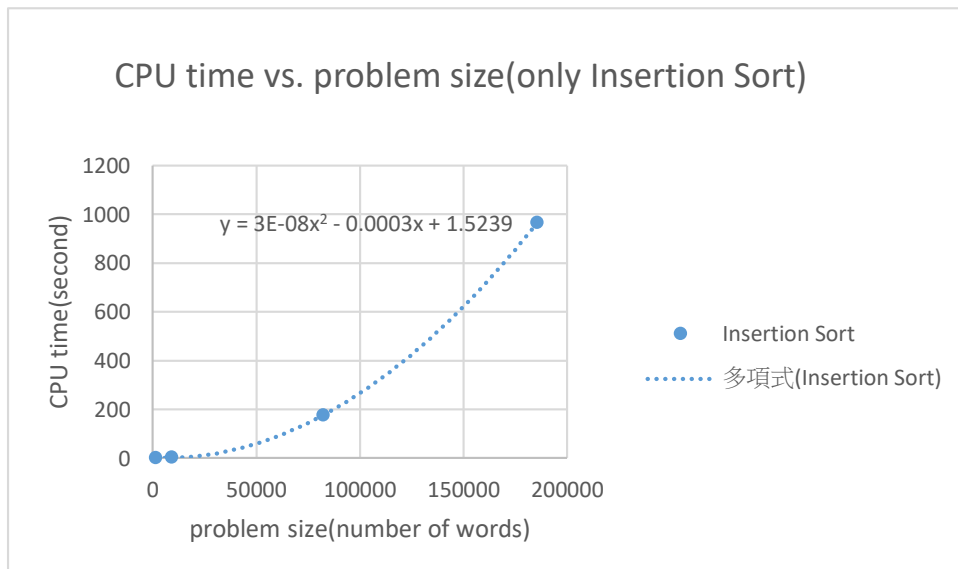


3. Memory usage vs. problem size



4. Bonus

1. Insertion sort 的趨勢線冪次為 2。



2. 由趨勢線可看出，merge sort, heap sort 和 quick sort 的冪次皆為 1，推測是測資量不夠的問題，如果測資量夠多，會接近 $n \log n$ 。

