

作業系統 Project1

一. 使用開發環境:

作業系統: Windows 10 家用版

處理器: Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz

系統類型: 64 位元

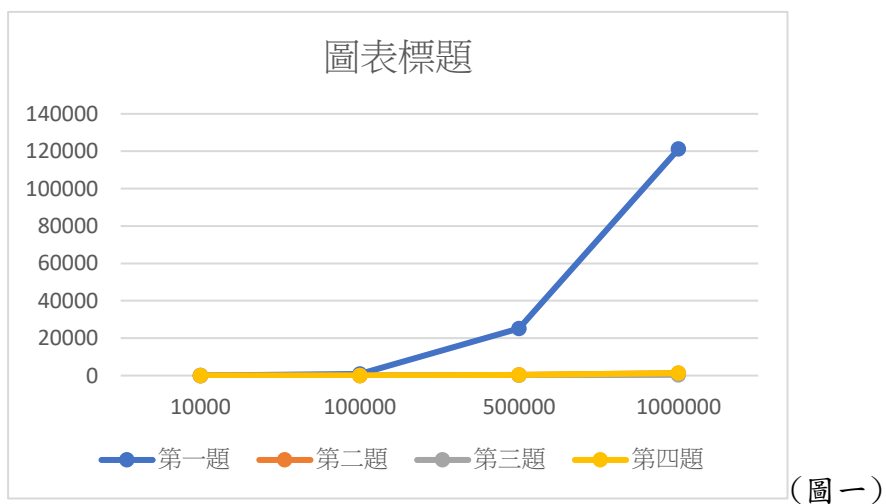
使用軟體: AnacondaNavigator → spyder

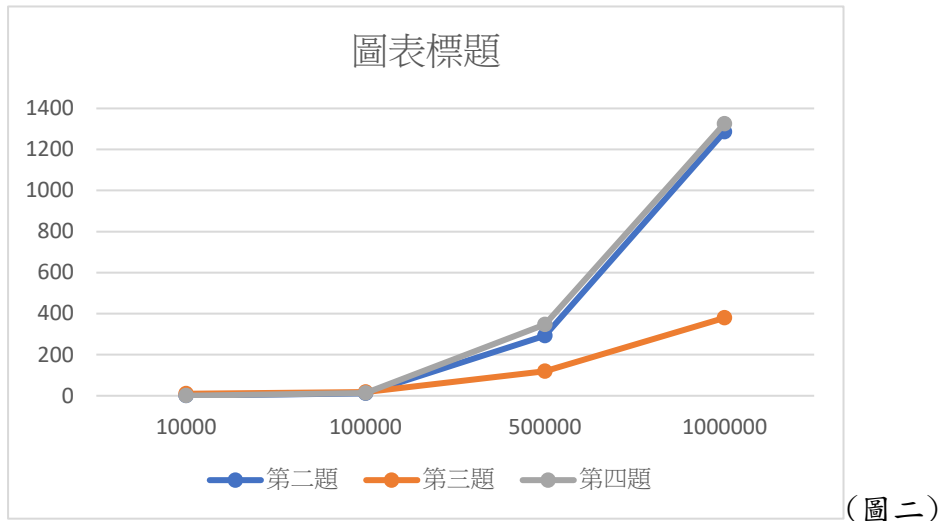
語言: python 3.7

二. 分析:

以 $K = 100$ 為例 (單位: 秒)

資料筆數	第一題	第二題	第三題	第四題
10000	10.2	0.297	9.819	0.237
100000	919.875	12.303	18.699	12.934
500000	25234.031	291.968	118.972	347.748
1000000	121187.672	1285.84375	379.3288	1325.443





根據以上的圖表圖一為一~四題的圖表，由此可以看出使用 python 作 Bubble Sort 跟 thread 與 multiprocessing 的時間真的相差太久，導致圖一看不出第二~四題的差異，所以我另外呈現圖二來表示第二~四題。

而由圖二來看，第三題的憤憤分了 K 個 Process 作 bubblesort 也比單純使用 K 個 thread 以及同一個 process 要來的快，不過在執行的時候，因為使用了 multiprocessing，所以 CPU 就幾乎沒有空閒的時間，由電腦的工作管理員可看出 CPU 使用率高達 95% 以上。

根據以上結果就能看出，因為 bubblesort 所要耗費的資源太大了，所以如果能將大量資料切成好幾份同時進行，並讓他們分別直接佔領 CPU，就能大幅加速整個城市的執行時間。

一. 執行結果

```
In [5]: runfile('C:/Users/Wang/Desktop/10620115/project1.py', wdir='C:/Users/Wang/Desktop/10620115')
```

Testing File : |

在 Testing File 處輸入檔案名稱(不須加入.txt)

```
In [1]: runfile('C:/Users/Wang/Desktop/10620115/project1.py', wdir='C:/Users/Wang/Desktop/10620115')
```

Testing File : input

3

Start K process Bubble sort and K-1 process Merge version...

K : |

以第三題為例，他會要你輸入 K 的值(要切幾分)

```
In [1]: runfile('C:/Users/Wang/Desktop/10620115/project1.py', wdir='C:/Users/Wang/Desktop/10620115')
```

Testing File : input

3

Start K process Bubble sort and K-1 process Merge version...

K : 10

執行時間 : 1.6102204000000029

輸入完後會印出執行結果，並產生排序後檔案 input_output.txt

```
input_output.txt - 記事本
檔案(E) 編輯(E) 格式(O) 檢視(V) 說明
9164 9166 9166 9167 9167 9168 9170 9170 9173 9173 9177 9179 9179 9180 9180 9182 9187 9189 9191 9191 9191 9191 9192 9193 9195 9196 9198 9198 9200
9201 9201 9202 9203 9203 9204 9204 9206 9208 9208 9209 9211 9211 9213 9214 9214 9215 9216 9216 9217 9218 9218 9219 9219 9219 9219 9219 9219
9221 9222 9223 9224 9225 9227 9228 9228 9229 9231 9231 9232 9234 9234 9234 9235 9235 9235 9238 9239 9240 9241 9241 9243 9245 9246 9247 9247
9248 9249 9249 9251 9254 9254 9255 9256 9257 9258 9259 9259 9262 9262 9263 9264 9266 9266 9268 9271 9271 9272 9273 9274 9275 9275 9276 9276
9276 9277 9278 9279 9281 9282 9282 9283 9284 9285 9285 9287 9288 9290 9291 9291 9292 9293 9294 9294 9295 9296 9296 9297 9298 9300 9301 9303
9306 9307 9307 9307 9308 9308 9308 9309 9309 9309 9310 9313 9313 9318 9319 9320 9320 9322 9322 9323 9324 9324 9325 9326 9326 9326 9328 9328
9328 9329 9329 9329 9329 9330 9330 9331 9333 9334 9335 9338 9341 9341 9342 9344 9346 9347 9348 9349 9352 9353 9354 9355 9355 9357 9358 9360
9360 9361 9361 9362 9362 9363 9366 9368 9368 9369 9372 9372 9373 9374 9374 9374 9376 9379 9379 9381 9383 9386 9386 9388 9388 9389 9390 9390
9391 9392 9392 9393 9393 9394 9395 9396 9397 9397 9400 9401 9402 9402 9402 9402 9402 9404 9404 9404 9405 9408 9410 9410 9411 9412 9412
9413 9413 9415 9416 9416 9424 9424 9426 9428 9428 9429 9429 9436 9439 9440 9441 9443 9444 9445 9446 9446 9448 9448 9450 9454 9454 9455 9456
9457 9457 9457 9462 9464 9467 9468 9468 9470 9471 9471 9472 9475 9476 9478 9480 9480 9481 9483 9484 9485 9488 9489 9489 9490 9491 9492
9494 9495 9495 9495 9495 9496 9496 9497 9497 9497 9500 9502 9502 9504 9504 9506 9506 9506 9509 9510 9512 9512 9512 9513 9514 9515 9516 9517
9518 9520 9524 9524 9526 9531 9531 9532 9532 9537 9538 9539 9540 9544 9545 9547 9552 9552 9553 9553 9554 9554 9556 9557 9558 9559 9560 9560
9560 9563 9565 9565 9567 9569 9569 9571 9573 9573 9573 9574 9574 9575 9576 9582 9584 9584 9588 9589 9590 9590 9591 9593 9594 9596 9597 9598
9598 9599 9600 9600 9600 9602 9603 9604 9605 9605 9608 9609 9610 9612 9616 9617 9618 9619 9622 9625 9625 9626 9626 9626 9626 9628 9630 9630
9630 9631 9631 9633 9633 9633 9633 9635 9635 9636 9636 9638 9638 9639 9639 9640 9641 9642 9643 9643 9646 9647 9647 9647 9648 9650 9651
9651 9654 9655 9657 9660 9660 9661 9661 9661 9662 9662 9663 9663 9664 9666 9666 9666 9667 9669 9670 9671 9671 9671 9672 9672 9672 9674 9675
9675 9675 9678 9683 9684 9685 9686 9687 9688 9689 9693 9693 9693 9694 9694 9696 9698 9698 9698 9701 9701 9703 9703 9706 9706 9707 9708 9710
9712 9713 9715 9718 9718 9718 9719 9720 9722 9722 9723 9723 9723 9725 9726 9728 9728 9729 9730 9735 9739 9739 9739 9742 9744 9745 9750 9753
9755 9757 9758 9758 9759 9759 9761 9762 9765 9765 9765 9766 9766 9767 9768 9769 9770 9770 9770 9772 9772 9774 9774 9777 9780 9780 9782 9782
9782 9782 9783 9783 9784 9786 9787 9789 9789 9790 9791 9791 9793 9793 9794 9794 9794 9796 9797 9798 9798 9799 9799 9801 9802 9805 9805 9805
9805 9808 9808 9811 9813 9815 9816 9816 9816 9819 9819 9823 9824 9827 9829 9829 9830 9831 9831 9832 9832 9833 9834 9835 9835 9835 9837 9837
9839 9839 9841 9841 9841 9845 9845 9846 9847 9847 9849 9854 9855 9857 9858 9858 9860 9861 9865 9865 9865 9866 9866 9867 9868 9869 9869 9871
9872 9872 9872 9873 9873 9876 9876 9877 9879 9880 9883 9883 9884 9885 9885 9886 9887 9887 9888 9889 9891 9892 9892 9893 9893 9894 9894 9896
9897 9897 9898 9899 9902 9904 9904 9906 9909 9911 9911 9912 9914 9916 9917 9917 9917 9919 9919 9921 9923 9925 9925 9925 9926 9927 9928 9928
9928 9929 9931 9932 9932 9933 9933 9934 9935 9939 9941 9942 9944 9948 9948 9949 9950 9952 9952 9953 9954 9954 9955 9956 9956 9957 9958 9959
9959 9960 9961 9961 9962 9966 9966 9966 9968 9968 9972 9972 9974 9975 9977 9977 9977 9978 9982 9985 9986 9986 9986 9987 9988 9988 9988
9988 9988 9989 9990 9990 9990 9993 9996 9997 9997 9998 10000
執行時間 : 1.61022 seconds
```

輸出檔案內容