

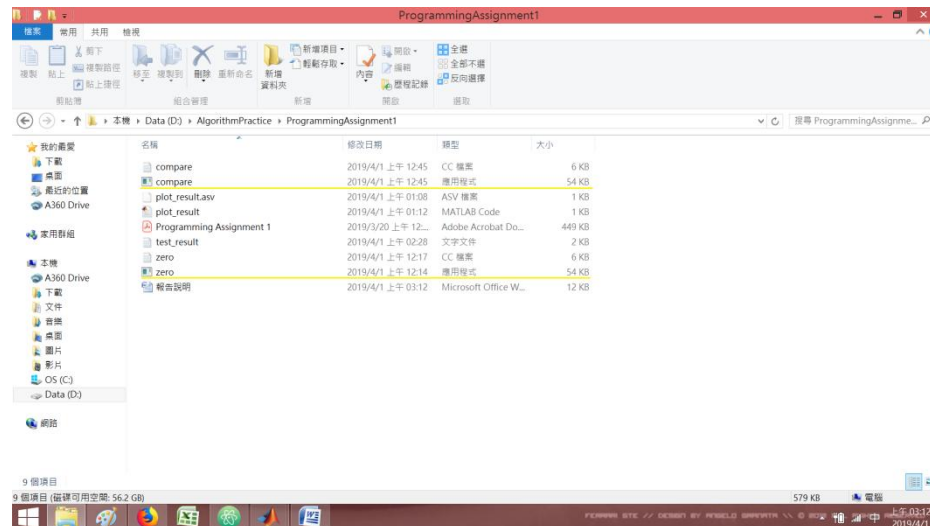
# Programming Assignment1

## Strassen's Algorithm

104303206 黃筱晴

## 1. 如何執行你的程式

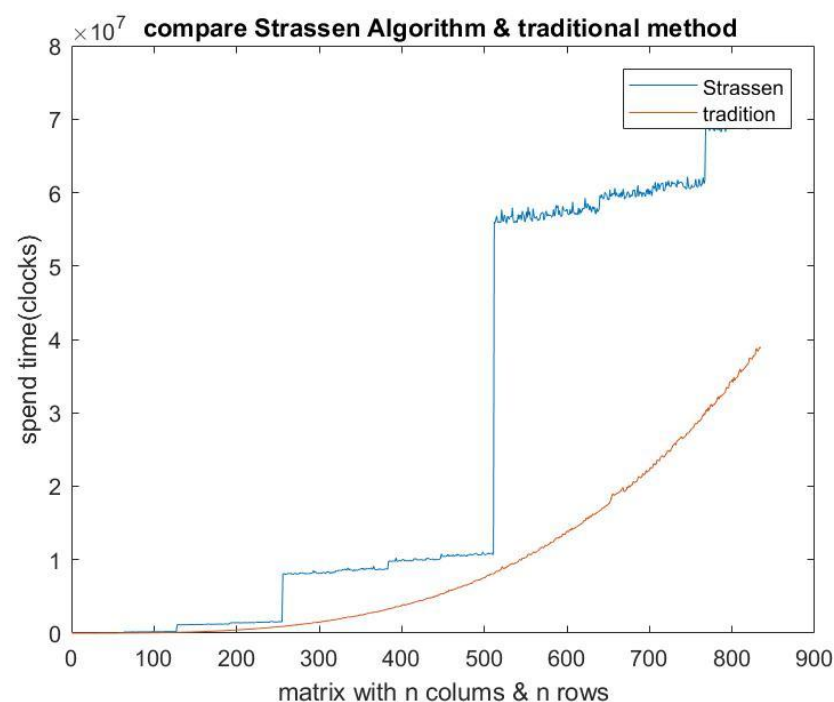
開啟檔案總管點擊 zero.exe 檔，或使用 cli 切換至該目錄輸入指令 `./zero` 測試第一題；開啟檔案總管點擊 compare.exe 檔，或使用 cli 切換至該目錄輸入指令 `./compare` 測試第二題。



## 2. 測資如何輸入

照著終端機顯示內容輸入，詳見說明影片內容。

## 3. 「使用傳統的矩陣相乘法」與「你實作的 Strassen's Algorithm」兩者完成矩陣相乘的時間比較。



如上圖，拉大尺度，Strassen 的曲線越來越近似對數函數。而傳統方法的曲線則大致上正比於  $n^3$  穩定上升。

使用 Strassen 演算法時，子問題的數目為 7，其執行時間滿足  $T(n)=7T(n/2)+O(n^2)$ ，此遞迴公式的解為  $T(n)=O(n^{\lg 7})$ ，優於傳統做法的  $O(n^3)$ 。理論上，當  $N$  夠大時 Strassen 演算法執行效率能超越傳統演算法。但是我的程式應該有某些問題以至於做不出來 QQ。

#### 4.問題討論

(1) 當輸入的 A、B 矩陣維度均不為  $2^n$  時，你的程式會進行何種處理？

補零，使他成為  $2^n * 2^n$  方陣。依據矩陣相乘的塊狀定理，此操作並不影響運算結果。

(2) 課本建議當程式遞迴到矩陣小到一個程度時，其實直接利用原本  $O(n^3)$  的矩陣相乘法即可，不必再用 Divide-and-Conquer 的概念切下去，則此門檻值大約是多少

正確數值與系統有很大的關係，需要以實驗來決定。很遺憾我的程式應該有點問題，這題沒測不出來 QQ。