**PL\_Group19\_Hw4**

**Compile :** gcc -g -Wall -fopenmp -o pl\_hw4 pl\_hw4.c

**Execute :** ./pl\_hw4 8 test3 output.txt

**使用語言 :** c

**使用方法 :**

1. **Traditional :**

使用openmp來做平行化的處理，input的第二個參數就是thread數目

1. **Strassen :**

使用strassen來做處理，並未進行平行化處理，但是有設計一旦切的大小達到256 \* 256便直接做一般的乘法，而不是繼續切，直到2 \* 2

**效能比較 :**

我們分別從1024 \* 1024 和4096 \* 4096來看。在1024 \* 1024的測資中，我們得到的結果是平行化處理過後的傳統乘法大約花了10秒，而Strassen只花了2.3秒；在4096 \* 4096中，我們得到的結果是平行化處理過後的傳統乘法大約花了30分鐘，而Strassen只花了3、4分鐘。(確實數據在Demo影片中有列出)

由此可見，Strassen的效率確實比起平行化的傳統乘法快了許多，不過這是在我們有對Strassen有做小小的優化的情況，如果是切到2\*2的話，那速度上就會慢了許多。

**組員分工 :**

Strassen’s multiplication：趙哲宏

傳統multiplication，影片製作：吳元智

整合兩份CODE：許力仁

投影片製作：郭家佑