

## 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

### 2. 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