

Programming Assignment 1

Strassen's Algorithm

2019.03.16

給定兩個 $n \times n$ 矩陣 A, B 要計算它們的乘積： $C = A * B$ ，最簡單的方式就是用講義上所提到的傳統的矩陣相乘法來做，但這需要花多少時間呢？

在 n 值只要夠大的條件下，則需要的時間複雜度可能會是 $\theta(n^3)$ 。所花費的時間成本與複雜度都是成正比的，這時候該如何改善呢？

已知目前有個號稱最快的矩陣相乘演算法，叫【Strassen's Algorithm】，它是利用 Divide-and-Conquer 的概念來進行矩陣相乘，所需要耗費的時間約為 $O(n^{\lg 7}) \approx O(n^{2.807})$ 。

但其終歸只是理論或經過數學證明，所以這次的作業就是要將【Strassen's Algorithm】實作出來，證明矩陣乘法確實可以在低複雜度的演算法中實作出來。

《題目》

我們知道當 n 值大到一定程度時，用 Strassen's Algorithm 的運算速度就會超越傳統的演算法，請自行設計測資並觀察實驗結果，矩陣 A, B 中的元素皆為整數，還要確保你實做出來的 Strassen's Algorithm 在 n 值大到一定程度的條件下還不會失敗，並於報告中回答下列兩個問題：(可以拍下觀察結果輔助說明)

1. 當輸入的 A, B 矩陣維度均不為 2 的倍數時，你的程式會進行何種處理？
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？

《小技巧》

同學可在每次做矩陣相乘之前，先將 A, B 矩陣的各元素做 mod 運算取餘數，以此方法讓元素數值變小，既可避免 integer 溢位的問題，也可以省下一些等待程式執行完成的時間，這邊規定請使用 **mod 101**，若你使用其他的數會使相乘出來的結果跟助教不同。使用此技巧不會影響整體的時間複雜度。

《報告內容》

報告需包含要如何執行你的程式、測資如何輸入、「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，以及上述兩

個題目。此份報告主要是讓助教了解你的解題思路及如何實驗並且得出結果，請同學盡可能的詳細描述。

《作業繳交及注意事項》

1. 可使用的程式語言為 C、C++、Java；需繳交你的程式之原始碼，原始碼中重要演算法部分需有註解，若使用專案則為整個專案檔。
2. 繳交一份報告，報告需包含的內容如上，Word 或 PDF 皆可
3. 如果你有其他可以說明你的作法的檔案（影片...等等），也可以一起繳交上傳。
4. 將所有繳交上傳的檔案，包成一個壓縮檔上傳至 LMS 作業區；檔名為「PA01_學號_姓名.(壓縮檔副檔名)」(ex. PA01_106502666_王小明.zip)，壓縮檔內至少要有 2 個檔案：程式碼 及 此次作業的報告說明。
5. 繳交期限：**04/02(二) 23:59**

《評分標準》

程式碼 80%、報告 20%，如有抄襲一律 0 分計算（包含抄襲網路）。

編譯不過、未實作出 Strassen's Algorithm：0 分

上傳檔名不符格式：扣 5 分