Matrix Multiplication

札記

1. 假設A = [a_{ij}]，B= [b_{ij}]皆為nxn的矩陣，欲計算這兩個矩陣相乘的結果C= [c_{ij}] = AB，若利用暴力法(brute force)計算，即直接用矩陣相乘的定義計算，其時間複雜度為$\Theta(n^3)$ (因為 $C_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}$， $\forall\ i \geq 1$ ，$j \leq n$， 假設 access 矩陣的每一項花費為$\Theta(1)$， 則計算一個c_{ij}需時$\Theta(n)$，共計有n^2個c_{ij}，因此時間複雜度為$\Theta(n^2 \cdot n)$)
2. 利用 Divide-and-conquer 設計演算法來改進矩陣相乘的效率，首先我們嘗試為將A與B個 切成4個大小為 $\frac{n}{2} \cdot \frac{n}{2}$ 的矩陣，利用這些子矩陣相乘的結果來得到C， 如下所示:
$$C = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B \end{bmatrix}$$
此法需要計算 $\frac{n}{2} \cdot \frac{n}{2}$ 大小的矩陣相乘共8次，因此共有8個子問題，每次遞迴計算出子問題的解後，執行矩陣的加法，需時$\Theta(n^2)$，所以若$T(n)$表示計算二個$n \cdot n$矩陣相乘的時間，則此法的時間複雜度為 $T(n) = 8T(\frac{n}{2}) + \Theta(n^2)$， 利用 master theorem 得 $T(n) = (n^3)$ ， 因此這樣的divide-and-conquer 效果並不會比暴力法要來的好。
3. 若能將2.中的子問題個數降低，應能提升 divide-and-conquer 的效能，著名的$Strassen's\ method$便是針對此點作改進，其psuedocode 可參考演算法2-4

Strassen's method

1. Divide:
$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$$

2. Conquer:
   $P = (A_{11} + A_{22})(B_{11} + B_{22})$
   $Q = (A_{21} + A_{22})B_{11}$
   $R = A_{11}(B_{12} - B_{22})$
   $S = A_{22}(B_{21} - B_{11})$
   $T = (A_{11} + A_{12})B_{22}$
   $U = (A_{21} - A_{11})(B_{11} + B_{12})$
   $V = (A_{12} - A_{22})(B_{21} + B_{22})$
3. Combine:
   $C = \begin{bmatrix} P+S-T+V & R+T \\ Q+S & P+R-Q+U \end{bmatrix}$

Time complexity of Strassen's method:假設T(n)表示計算二個nxn矩陣相乘的時間，因為在計算 P, Q, R, S, T, U, V這7個子矩陣中共需遞迴計$\frac{n}{2}\cdot\frac{n}{2}$大小的矩陣相乘共7次，所以會產生7個子問題，其餘若干次$\frac{n}{2}\cdot\frac{n}{2}$大小矩陣相加皆需時$\Theta(n^2)$，所以$T(n)$的遞迴關係式為

$$\begin{cases} T(n) = 7T\left(\frac{n}{2}\right) + \Theta(n^2) \\ T(1) = \Theta(1) \end{cases}$$

利用master theorem分析得$T(n) = \Theta(n^{lg7}) \approx \Theta(n^{2.81}$