時間複雜度

原講義作者：林易逵

吳浩瑋

課程大綱

- 什麼是時間複雜度?
- 時間複雜度對我們來說有什麼用?
- 練習估計時間複雜度的例題

什麼是時間複雜度?

- 一個描述程式運行所需時間的函式
- 習慣上假設所有基本操作所花費的時間都是一單位(事實上各運算花費時間有著常數倍的落差)
- 對於我們而言，我們比較在乎時間複雜度與處理資料量的關係
- 舉例?

時間複雜度與處理資料量的關係-舉例

- A陣列中有N筆資料，求出每一對$A_i \times A_j \ (1 \leq i < j \leq N)$的和
- 總共有幾對i, j?
- 如果直接窮舉的話，整體時間複雜度是?
- 時間複雜度$T(N) = \dfrac{N(N+1)}{2} = \dfrac{N^2}{2} + \dfrac{N}{2}$
- 這樣表示會不會太複雜?

Big-O

- 念作O-of
- 當f(x)=O(g(x))時，代表存在一常數M，使得$M|g(x)| > |f(x)|$
- 白話翻譯：我們只在乎最高次項，係數和低次項忽略
- 舉例
- $O(2N+2) = O(N)$
- $O(N^2/2 + N/2) = O(N^2)$
- $O(2^N + N^2 + 1) = O(2^N)$

時間複雜度對我們來說有什麼用?

- 很有用!!!
- 可以估計你的程式需要跑多久，讓你不會再得到TLE
- 我們通常在意的是最糟情況下的複雜度

估計出來的時間複雜度代表什麼?

- 一般的OJ大概1秒可以跑10^8個指令(當然每個OJ狀況不同，不過會介於10^7~10^10之間
- 如果說複雜度只跟N有關的話，當題目說：
- $N \leq 10 \rightarrow O(N!), O(4^N)...$
- $N \leq 20 \rightarrow O(2^N)...$
- $N \leq 50 \rightarrow O(2^{\lfloor N/2 \rfloor}), O(N^4)...$
- $N \leq 500 \rightarrow O(N^3)...$
- $N \leq 5000 \rightarrow O(N^2)...$

估計出來的時間複雜度代表什麼?

- N≤5×10^4→O(N^1.5)…
- N≤10^5→O(N^1.5),O(NlogN)…
- N≤10^6→O(NlogN),O(N)…
- N≤10^7→O(N)…
- N≤10^12→O(√N)…
- N≤10^18→O(logN)…

再回去看剛剛的例題

- •A陣列中有N筆資料，求出每一對Ai×Aj(1<=i<j<=N)的和，N≤〖10〗^6，時限10秒
- •這次有給範圍了，我們剛剛已經有了O(N^2)的解，可是這樣夠快嗎?
- •O(N^2)的解帶入N≤10^6 要跑10^12次， 10^12/10^8 =10000秒，還不夠快!
- •那這題我們要怎麼做呢?