**時間複雜度 — 遞迴(下) — Master Theorem**

本篇接續[上一篇](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394)，要介紹的是第二個方法Master Method。

**本篇章節：**

*1. Master Theorem定義*

*2. Master Theorem所出現的三種 Case*

*3. 幫助理解原理之 Recursion Tree*

*4. Master Theorem的出處*

*5. 總結論*

*6. 範例*

**分析Divide & Conquer演算法時，我們通常使用以下兩種方法：**

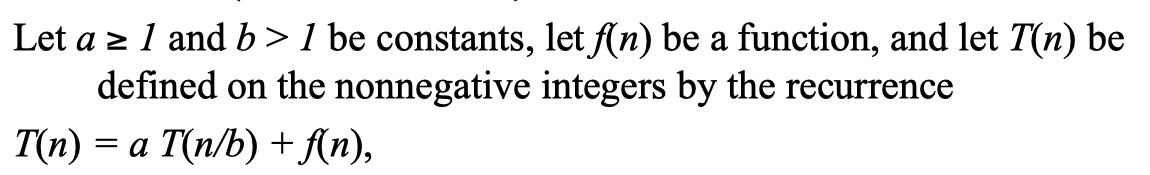
1. Recursion-Tree Method搭配Substitution Method(這裡稱為「數學歸納法」)

2. Master Method：這個方法很直觀（有時間的話，建議先瞭解「方法一」再了解這個方法會比較全面）

**為什麼會說Master Method比較直觀呢？**

*Ans: 因為它就像食譜(Cookbook)一樣，只要照著書上的指示，就可以找到得到美味的食物(答案)。*

**1. Master Theorem定義：**



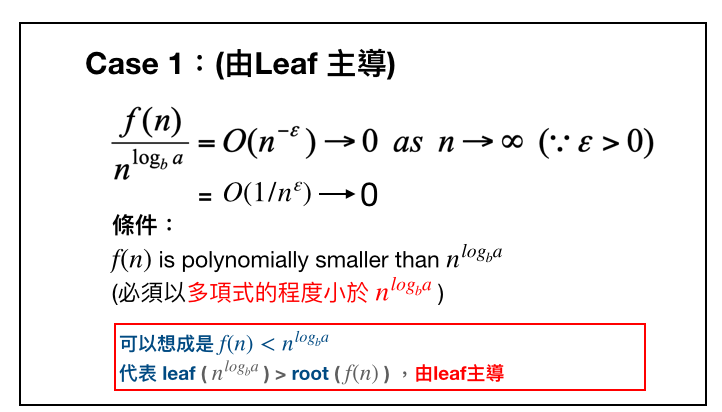
以我自己理解後的翻譯：

假設有個 a ≥ 1和 b > 1 的常數，f(n)為一函式，然後假設 T(n)定義在非負整數上，遞迴公式如下：T(n) = a T( n / b ) + f(n)。

**2. Master Theorem所歸納的三種情況：**

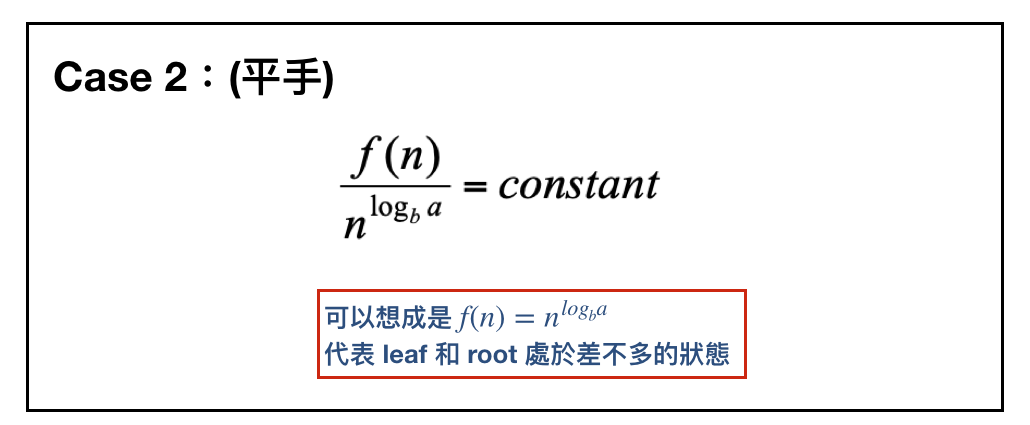
依照上述定理，歸類出三種情況（這邊先講結論之後，之後再說明理由）

Case 1：



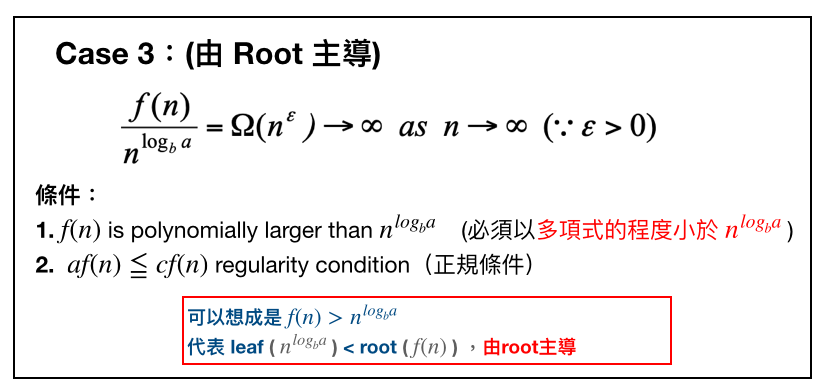
Case 1

Case 2：



Case2

Case 3：



Case3

**為什麼Case 3 要多加一個 Regularity Condition（正規條件）?**

*Ans: 只是想要更加確認，所得出的答案一定會贏過「葉節點的部分」。*

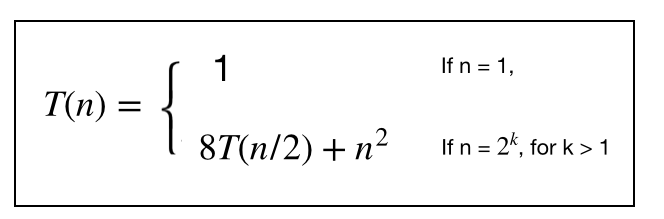
*詳細說明請參考：*[*Regularity condition*](https://cs.stackexchange.com/questions/4854/why-is-there-the-regularity-condition-in-the-master-theorem)

在上述所說的3個Case中，應該會想說為什麼都是用log跟f(n)來做比較吧。接下來就讓來做解釋吧！

**3. 幫助理解之Recursion Tree**

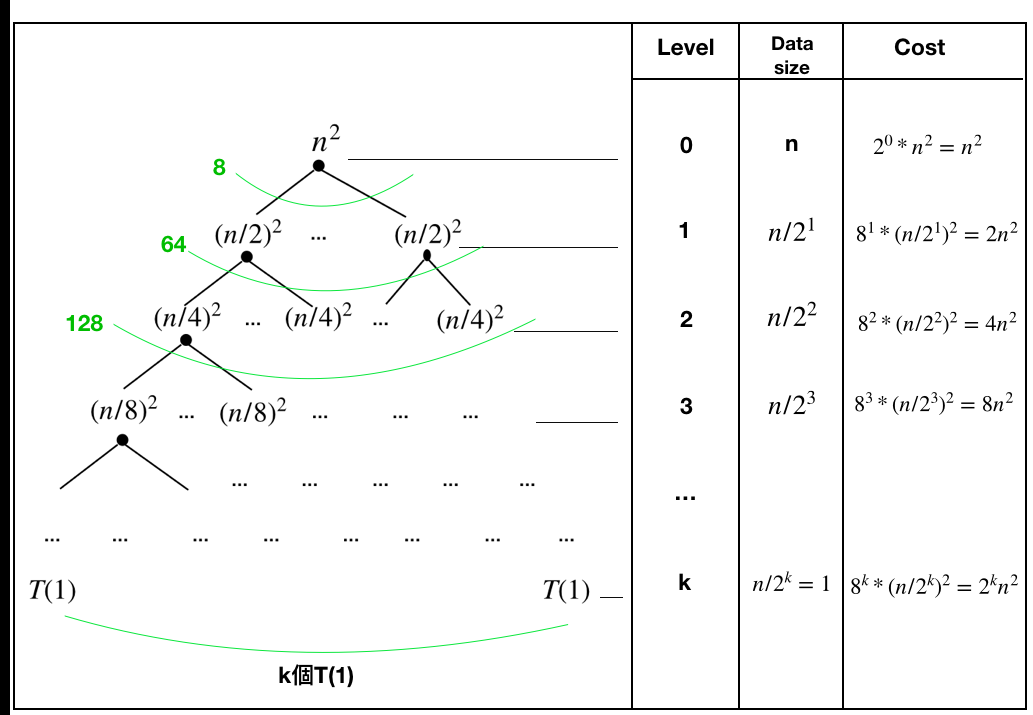
先看一個範例來幫助後面的理解。

遞迴關係式如下：



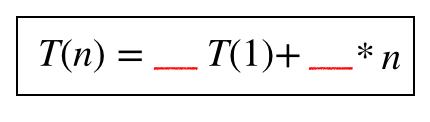
圖一

畫 Tree應該要逐步來畫，為了方便，這邊暫時省略。



圖二

有了上面的圖後，我們可以用[上一篇](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394)所說的來將此遞迴關係式作拆解。

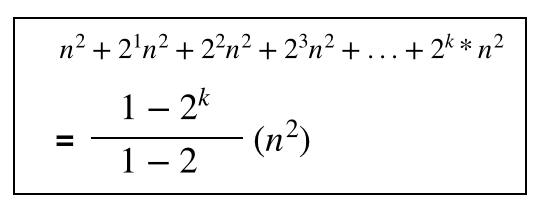


圖三

這邊跟上次有c \* n 的不一樣是因為，上次題目給的遞迴關係式有Θ(n)，為了做替換，所以我們把它換成cn。而這次題目沒有給，直接給了n。

**首先，先看我們Cost的部分，總共做了多少次的合併？**

*Ans: 從圖二，看到每一層的 Cost從 n² + 2¹n² + 2²n² + 2³n³ +… + 2^k n² ，先把 n²提出來，再利用等比公式得到：*

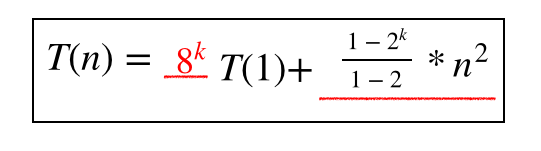


圖四

**再來，前面T(1)的部分，要得出T(1)的話，前面需要拆成多少個子問題？**

*Ans: 從圖二得到，經過 8^k次方後，我們可以拆解到 T(1)。*

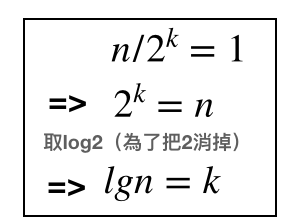
總和上述結果：



圖五

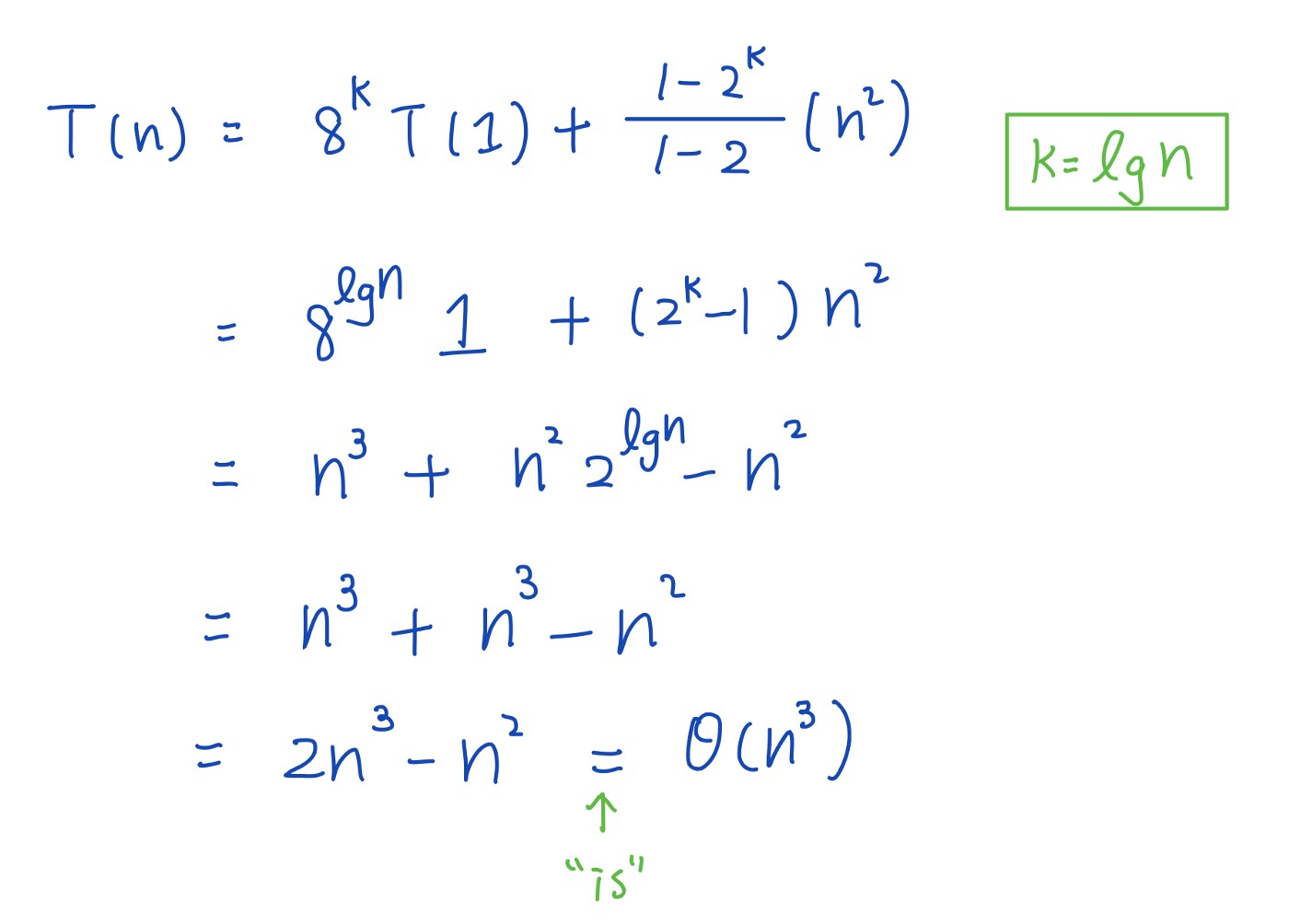
**那k又是多少？**

*和上一篇所說的相同，從圖二得到：****n/ (2^k) = 1****(原始資料 n不斷砍半再砍半，經過 k次的砍半後，最終只剩一個)*



圖六

整合所有的資訊後：



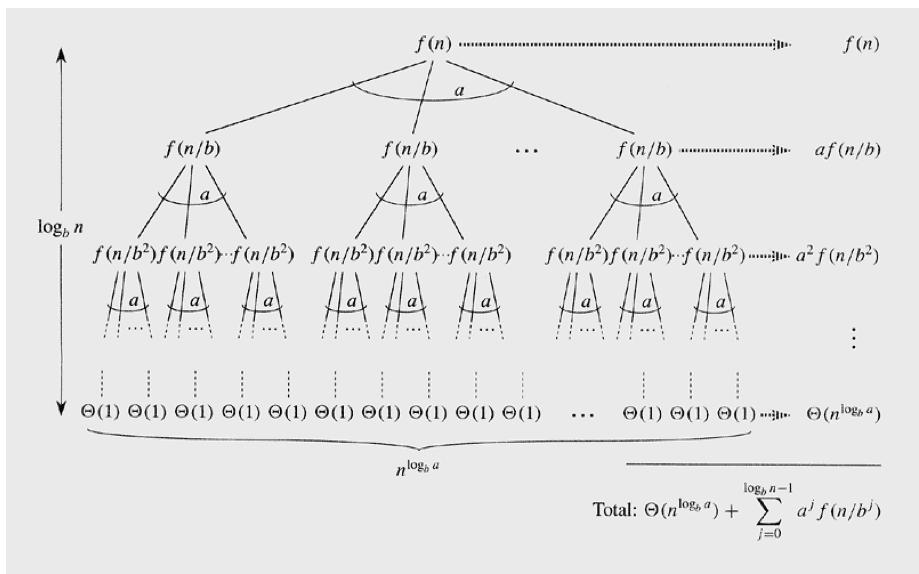
圖七

想必看到這邊一定會有疑問，那這棵樹到底跟Master Theorem的logb a有什麼關係啊？

我們就快要講到了！再等一下～～

**4. Master Theorem的出處**

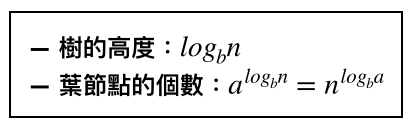
由前面的範例後，我們可以自然的延伸到下圖：（這邊我就直接把參考資料的圖片，放上來）



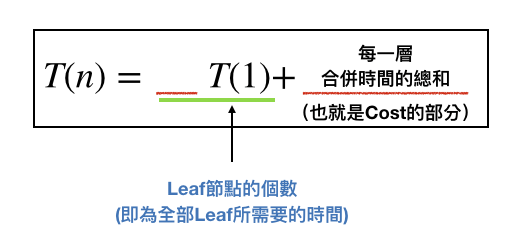
圖八

看到上面一對可怕的數字，大家不用害怕，先從圖上面找一些性質。

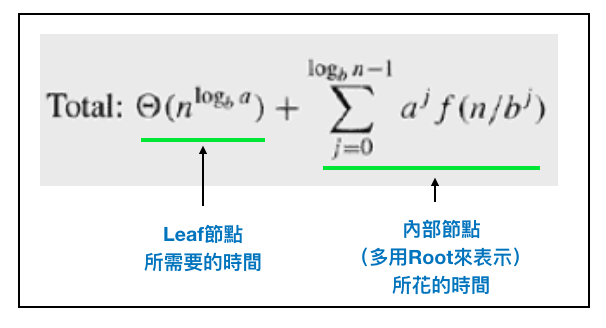
**從圖八中Recursion Tree所得到的特性：**



把圖八中的 Total ，類似此式：

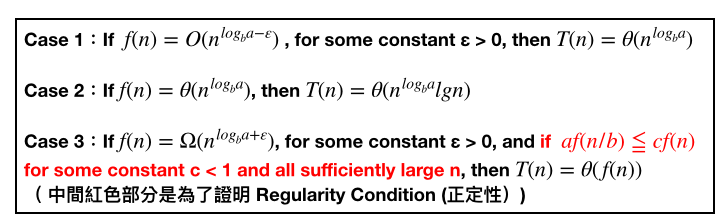


最後我們用上面的想法來帶入課本的結論得到：



**其實這些遞迴關係式即是把他們拆成兩個部分，一個是由leaf主導，一個是由root主導。用「主導(Dominated)」這個詞的原因是說，用這兩個來判斷，是Leaf還是Root影響T(n)比較多，如果較多的那個理所當然，漸進函數就會以它為主了。**

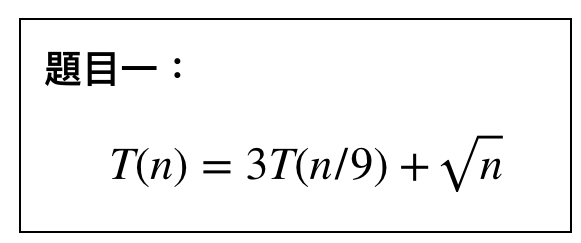
**5. 總結論**

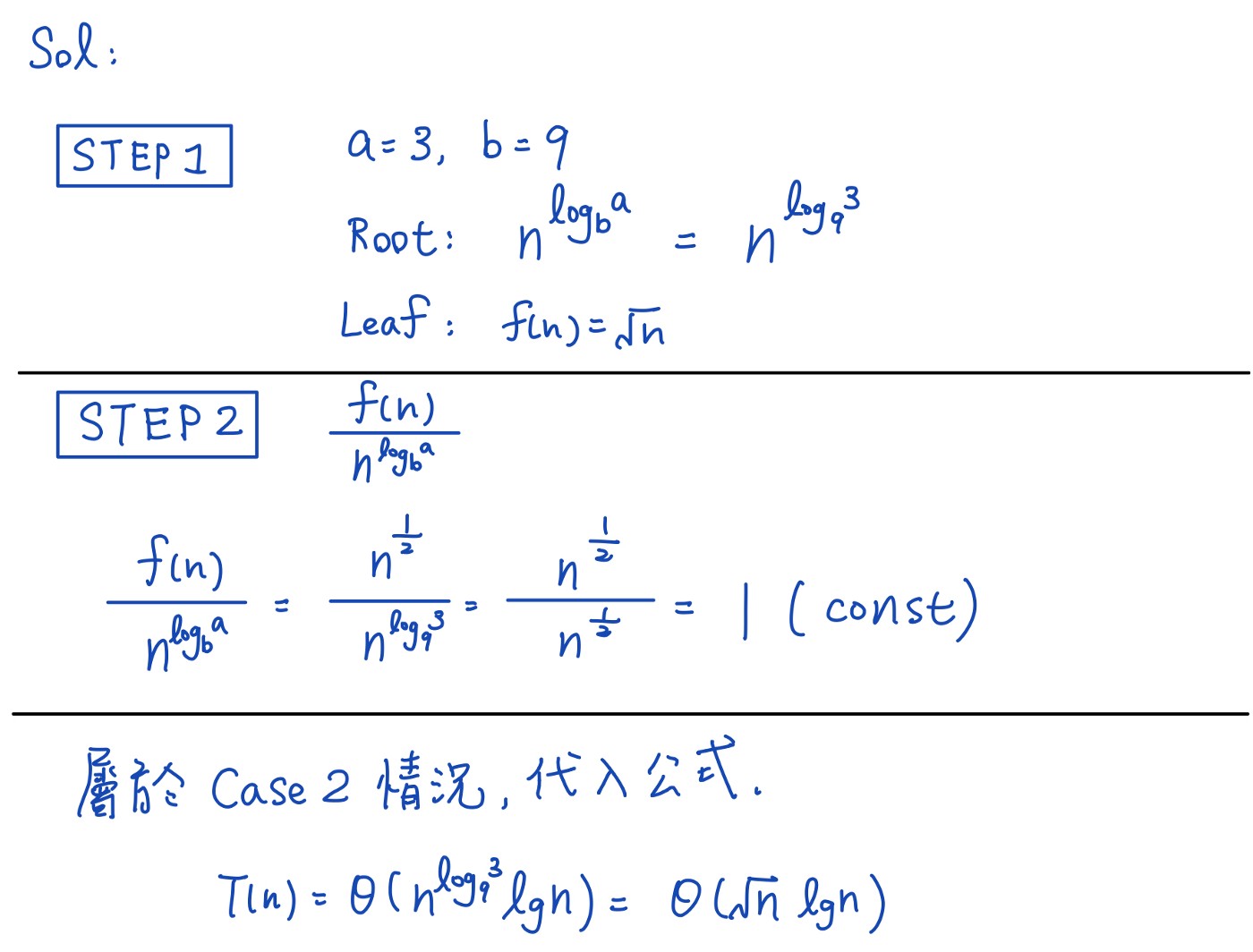


如果不清楚怎麼用的話，請看下方例子。

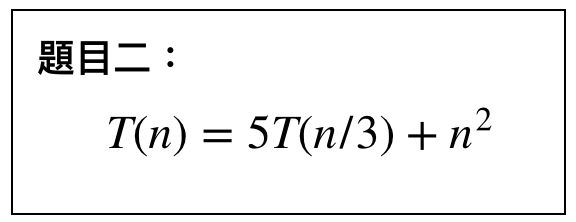
**6. 範例**

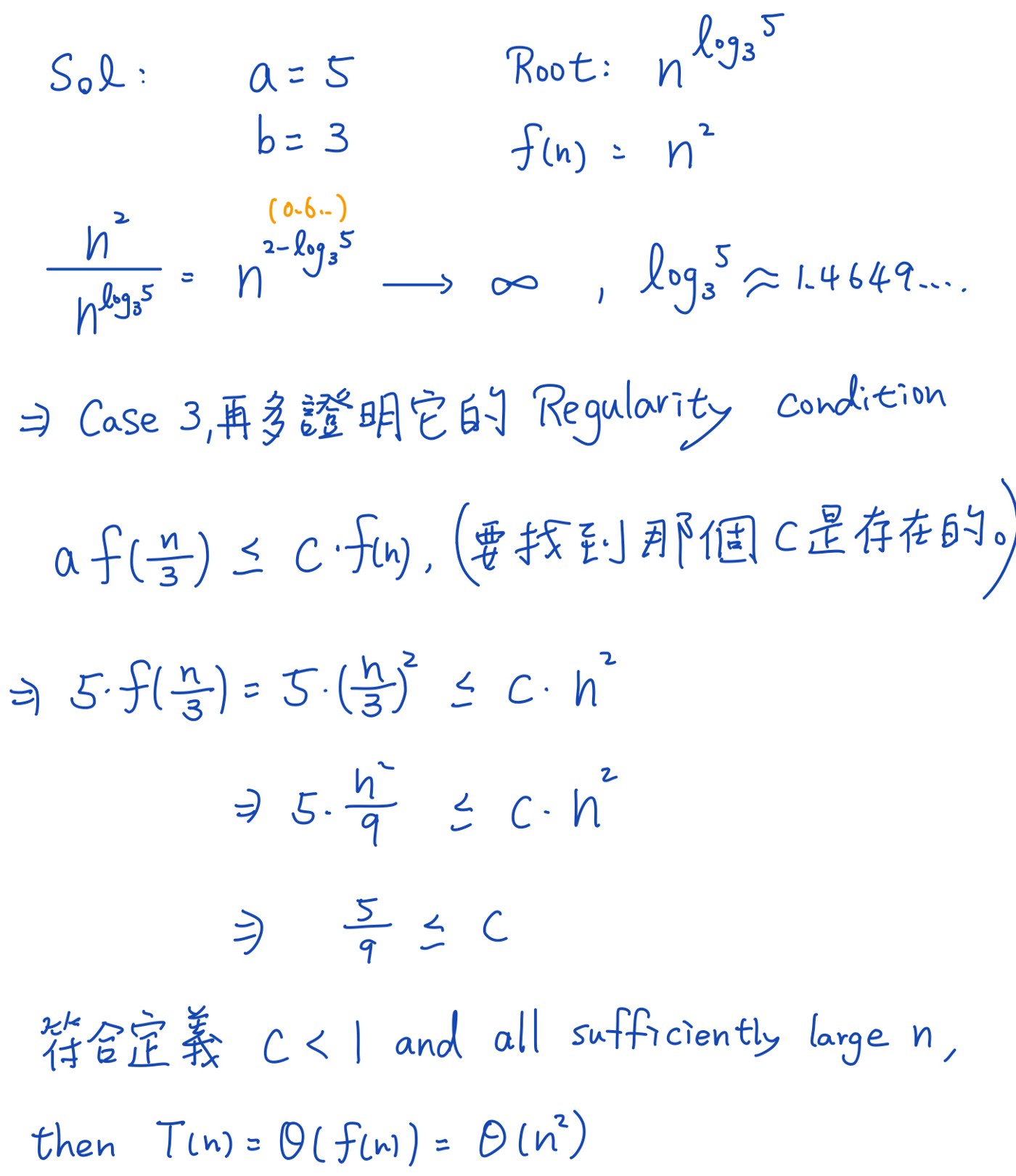
題目一：





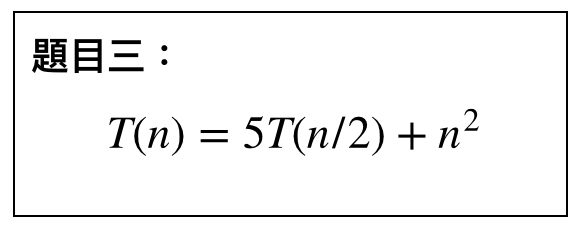
題目二：

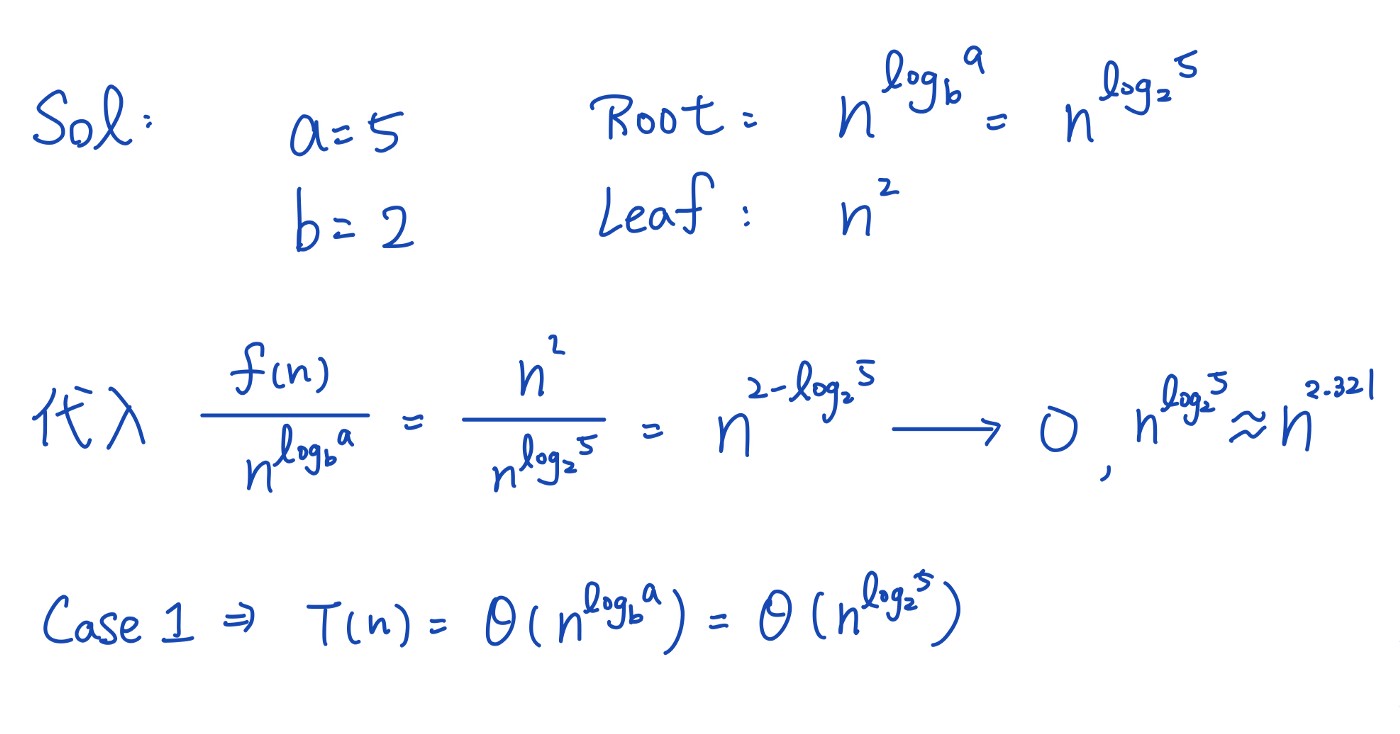




**Case 3比較麻煩一點，還需要多證明一個Regularity Condition**

題目三：





這篇也就到這邊結束了，如果有錯誤還請各位多多指教，比較多手寫的部分，希望大家不要太介意。那我們就下次再見吧～～

70

70

[More from Sharon Peng](https://mycollegenotebook.medium.com/?source=post_page-----307ad4608ab6--------------------------------)

Follow

一起精進程式能力吧！！

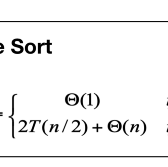
[Apr 30, 2021](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394?source=post_page-----307ad4608ab6----0----------------------------)

**[時間複雜度 —遞迴（上）](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394?source=post_page-----307ad4608ab6----0----------------------------)**

[這次要來討論的是 — Divide and Conquer(中文俗稱「分治法」），主要是把一個大大的問題，分成很多小問題，並一一破解他！ 在解相關的問題時，其時間複雜度，通常都使用遞迴關係式來表達，所以這篇要帶大家來找divide and conquer的時間複雜度關係，也就是當我今天寫出一個divide and Conquer的演算法(可以想成Mer …](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394?source=post_page-----307ad4608ab6----0----------------------------)

[Time Complexity](https://medium.com/tag/time-complexity?source=post_page-----307ad4608ab6---------------time_complexity-----------------)

[6 min read](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394?source=post_page-----307ad4608ab6----0----------------------------)

[[](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394?source=post_page-----307ad4608ab6----0----------------------------)](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E9%81%9E%E8%BF%B4-%E4%B8%8A-f6d51a462394?source=post_page-----307ad4608ab6----0----------------------------)

Share your ideas with millions of readers.

[Write on Medium](https://medium.com/new-story?source=post_page_footer_cta_write-------------------------------------)

[Apr 29, 2021](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E5%87%BD%E6%95%B8%E9%96%93%E7%9A%84%E6%AF%94%E8%BC%83-3e9ce0788f0f?source=post_page-----307ad4608ab6----1----------------------------)

**[時間複雜度 — 函數間的比較](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E5%87%BD%E6%95%B8%E9%96%93%E7%9A%84%E6%AF%94%E8%BC%83-3e9ce0788f0f?source=post_page-----307ad4608ab6----1----------------------------)**

[希望上次所提到的內容，各位讀者都有暸解到其中的精髓。 1. 幫助函數之間比較 — 數學公式與技巧 2. 簡單的例題（不同函數之間的時間複雜度的比較） 分析函數間的關係，目的在於要學會知道自己寫出來的程式，是不是真的可以使用（不會讓使用者等太久），或是當有兩個人用程式解決相同問題，老闆要怎麼知道誰的比較好，所以要靠今天的章節，來證明「我的演算法比較好！」。 而函數間的比較，我們可以想成是兩個函數在比賽，看哪一個成長速度會超過另一個。 1. 幫助比較函數間關係的簡單公式（方法一） 這邊取極限是在比較 f(n), g(n)的關係，所以 — 出現無限代表，f(n)贏了g(n)，因為f(n)跑得夠快，g(n)根本來不及追上，也可以想成說 g(n)根本來不及把 f(n)約掉，f(n)就直接往上衝（以圖形的角度來看）。 — 出現 0代表，代表g(n)贏了f(n)，因為 f(n)跑得很慢，g(n)衝太快，也可以想成說 g(n)很快就把 f(n)的值約掉，f(n)根本來不及增長。](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E5%87%BD%E6%95%B8%E9%96%93%E7%9A%84%E6%AF%94%E8%BC%83-3e9ce0788f0f?source=post_page-----307ad4608ab6----1----------------------------)

[Time Complexity](https://medium.com/tag/time-complexity?source=post_page-----307ad4608ab6---------------time_complexity-----------------)

[5 min read](https://mycollegenotebook.medium.com/%E6%99%82%E9%96%93%E8%A4%87%E9%9B%9C%E5%BA%A6-%E5%87%BD%E6%95%B8%E9%96%93%E7%9A%84%E6%AF%94%E8%BC%83-3e9ce0788f0f?source=post_page-----307ad4608ab6----1----------------------------)