

# 遞迴演算法簡介

許介彥

大葉大學 通訊與計算機工程學系

例一：「從前，有一個和尚作夢夢到一個和尚在作夢，這個夢中的和尚又夢到一個和尚在作夢，這個夢中的和尚又夢到一個和尚在作夢」

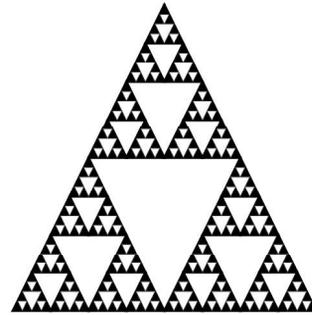
例二：「有一個畫家坐在窗前畫畫，畫中是一個畫家坐在窗前畫畫，畫中是一個畫家坐在窗前畫畫」

以上兩個例子有什麼共同的特徵嗎？很明顯，它們都是講到某件事情發生的過程中，又包含了與事情本身很類似的另一件事情，而且這種包含關係可以無止盡地發展下去；儘管這樣的情境在日常生活中並不多見，類似的概念卻發展出一個在電腦程式的寫作上非常重要的技巧，稱為遞迴（recursion）。

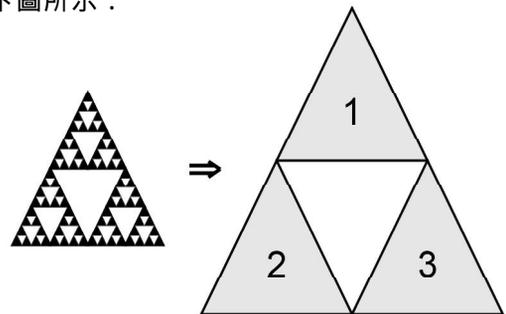
## 什麼是「遞迴」？

當我們說一個「東西」具有遞迴的性質，指的是這個東西的內部包含了與這個東西本身的形狀或性質類似的東西，而且這種包含關係可以一層一層地發展下去；也就是說，大東西中包含了小東西，小東西中包含了更小的東西，更小的東西中又包含了更小的東西，而且所有這些東西除了在大小上有差別外，在形狀或概念上其實全都非常相似。

下圖是遞迴的概念展現在圖形中的一個例子：



上圖可以看成是由三個形狀與原圖類似但是大小僅為原圖四分之一的小圖組成，如下圖所示：



當然，每個這種大小為原圖四分之一的小圖又可以看成是由三個形狀與原圖類似但是大小都僅為原圖十六分之一的小圖組成。數學上，這類含有遞迴概念的幾何圖形稱為碎形（fractals），近年來在學術界受到了廣泛的重視。

遞迴的概念除了可以用來畫圖，更廣泛地說，它可以是一種做事的方式，可以是我們在日常生活中完成某些事情的方法，也可以被應用到電腦程式中，讓電腦以遞迴的方式完成某些工作（所寫成的程式稱為「遞迴程

式」)；許多常見的資料結構可以用遞迴的方式來定義，許多程式語言的語法也常用遞迴的方式來表達；在數學上，遞迴的概念可以用來定義函數、數列、集合等，在許多場合都有重要的應用。

### 一個日常生活中的例子

假設你參加了某個慈善團體，某天，這個團體分派給你一個任務，要你為他們募款 1000 元；為了避免造成捐獻者太大的負擔，該團體規定每位捐獻者只能捐一元，因此你必須要找到 1000 名捐獻者。你會怎樣完成這項任務呢？

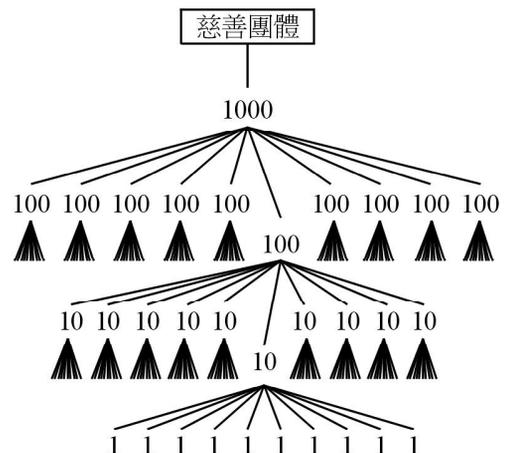
你可以採用非常直接的做法，發動你的親朋好友大家一起共襄盛舉，每個人捐出一塊錢；如果人數不足 1000 人，你可能會找個人潮洶湧的街頭，擺放一些看板或海報來勸募，直到湊足 1000 人為止，然後將募得的 1000 元交回給該慈善團體。這個做法當然可行，不過你會非常辛苦。

如果我們用程式的觀點來看上面的做法，可以將慈善團體看成是一個主程式，它交付給你一項任務，希望你在達成後向它回報，因此你就像是一個被它呼叫的副程式。如果我們將這個副程式取名為 COLLECT，它的工作可以用以下較接近演算法的方式表達出來：

```
COLLECT (n)
  for i ← 1 to n
    請某人捐出一塊錢
  交回所有募得的錢
```

當然，主程式在呼叫此副程式時，傳給它的參數（即副程式中的  $n$  的值）是 1000。

讓我們考慮以下另一種做法：你找來 10 個你的親朋好友，請他們每人負責募款 100 元，並將募款所得交回來給你，這樣一來你就同樣也有 1000 元可以交差了。這個做法對你來說要比上個做法輕鬆許多，你的工作只是找到 10 個人，將工作分配出去，接著就待在家裡等著收齊別人募來的款項就行了。當然，被你找到的 10 個人也可以依法炮製，每個人另外再找 10 個人，請每個人負責募款 100 元，這些被找到的人只要每人另外再找 10 個人，要求每個人捐出一元，金額總數將為 1000 元。下圖顯示了整個運作的過程：



這個做法只有最下層的 1000 人才是真正捐錢的人，所有其他人的工作有一個共通之處，就是都可以看成是分為「接受指派金額、將工作分配給 10 個人、將 10 個人繳回的錢收齊、將錢交給上級」等四個步驟；如果我們用程式的眼光來看，將每個人所做的工作看成是一個副程式的執行，那麼除了最下層的 1000 人以外，其他人的工作其實都可以用下面的副程式表示出來：

```
COLLECT(n)
  for i←1 to 10
    請某人負責募得(n / 10)元
    將 10 個人繳回的錢收齊
    將所有募得的錢交給上級
```

能不能將這個的副程式改寫一下，使得它也可以適用於最下層的 1000 個人呢？答案是肯定的。與較上層的人的工作相比，最下層的人的工作還是保有「接受指派金額、將錢交給上級」等兩個步驟，只是因為所須負責的金額只有一元，已經小到不能再細分了，他們的工作因此只要直接將一元交給他們的上級就好了。有了這樣的認識，我們可以寫出一個適用於全部的人的副程式：

```
COLLECT(n)
  if n= 1
    交一元給上級
  else
    for i←1 to 10
      請某人負責募得(n / 10)元
      將 10 個人繳回的錢收齊
      將所有募得的錢交給上級
```

這個副程式適用於所有的人，當我們需要某人負責繳回  $n$  元時，就可以呼叫 COLLECT( $n$ )來達成，不管  $n$  是 1000、100、10、或 1 都適用，既然如此，除了最下層的 1000 個人以外，較上層的每個人都可以透過呼叫此副程式 10 次來將工作分配出去；因此我們又可以將上面的副程式改寫如下：

```
COLLECT(n)
  if n = 1
    交回一元
  else
    for i←1 to 10
      COLLECT(n / 10)
    將 10 個人繳回的錢收齊
    將所有募得的錢交給上級
```

上面的 COLLECT 副程式最大的特色在於當傳入的  $n$  不等於 1 時，它在執行的過程中會呼叫 COLLECT 副程式本身 10 次。一個執行過程中會呼叫到本身的程式稱為「遞迴程式」( recursive program )，而遞迴程式所依循的演算法稱為「遞迴演算法」( recursive algorithm )。

請讀者特別留意上面的做法中，任務的分派是由上而下，而成果的累積卻是由下而上；要完成「募款 1000 元」的任務，必須先完成 10 個「募款 100 元」的任務；要完成每個「募款 100 元」的任務，必須先完成 10 個「募款 10 元」的任務；要完成每個「募款 10 元」的任務，又必須先完成 10 個「募款一元」的任務。

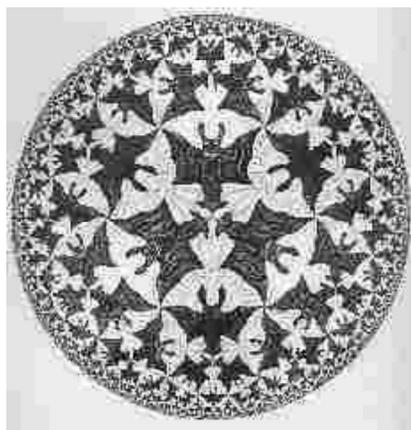
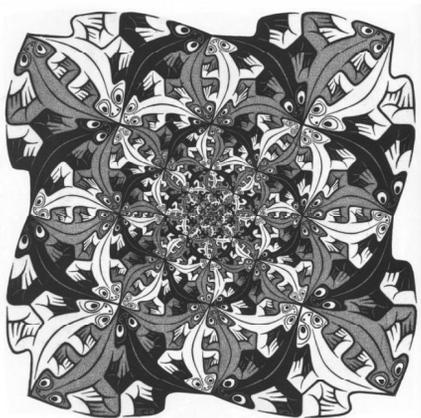
### 結語

遞迴的做法其實是一種「大事化小」的做法，當我們要用遞迴的方式解決某個問題或完成某項工作（如上例的募款  $n$  元）時，我們是透過解決與原來的問題類似但較簡單的問題來解決原來的較難的問題。上例中，我們要解決的是募款問題，我們是透過 10 個「募款  $(n / 10)$  元」的工作來解決「募款  $n$  元」的工作；募款  $(n / 10)$  元與募款  $n$  元當然是類似的工作（都是籌募某個款項），而所要募得的金額越小當然越容易達成。

針對某項工作，如果我們已經知道如何透過解決較簡單的情形來解決較難的情形，而且可以依此類推，用更簡單的問題繼續將問題簡化，所要解決的問題終將被簡化到已經簡單得無需再簡化或不能再簡化的地步，

這時候我們就可以很順利地將非常簡單的問題解決，然後循著之前將問題簡化的過程往回走，一路解決越來越難的問題，終於將原來的問題解決。

除了在數學及程式方面的應用外，遞迴的概念也常被許多藝術家用於藝術作品的創作上，不管是音樂、美術、建築、彫刻等方面都衍生出許多有趣的作品。下面是荷蘭著名藝術家 Maurits Cornelis Escher (1898-1972) 的兩幅作品，讀者不難體會其中隱含的遞迴的概念。



### 參考資料

- 1.許介彥 (2000) 淺談「計算機演算法」, 科學教育月刊, 第 235 期。
- 2.許介彥 (2001) C 語言迷魂陣 - 剖析 C 語言, 文魁資訊。
- 3.Eric S. Roberts, The Art and Science of C: An Introduction to Computer Science, Addison-Wesley, 1995.
- 4.Niklaus Wirth, Algorithms + Data Structures = Programs, Prentice-Hall, 1976.  
作者信箱 : chsu@mail.dyu.edu.tw

---

(上承第 55 頁)

- (1)本期有五題徵答題，請照「中學生數學通訊解題答題規則」中的規定作答。(參閱師大科學教育月刊 226 期)
- (2)本期徵答題不限您作答的題數，請於 91 年 1 月 1 日前將回函寄達：(100)台北市南海路 56 號，台北市立建國高級中學，楊希聰老師收。(信封上請註明通訊解題)
- (3)徵答題可能有多種解法，本期參考答案與

徵答者之優良解答，答題優良者姓名、就讀學校，將於 91 年 2 月份在台灣師範大學科學教育月刊及建國高級中學數學科網站上發佈。

(4)進入建中網站方法：

- 1.先利用瀏覽器進入建中首頁(網址：<http://www.ck.tp.edu.tw/>)
- 2.至最新消息點選數學科通訊解題。