

投稿類別：資訊類

篇名：

老少咸宜的益智遊戲---數獨

作者：

李亞承。國立瑞芳高級工業職業學校。資訊三

指導老師：

陳思亮主任

## 壹●前言

### 一、研究動機

第一次接觸數獨，是偶然在報紙上發現的，起初我認為這不過是個普通的小遊戲，便信手拾筆試著解答，然而它卻出乎意料地困難！至此數獨的魔力開始縈繞在我的心中，激發我的求知慾與好勝心，令我仔細去思考推敲：到底要如何解開這個題目！於是，我繼續拿起鉛筆和橡皮擦與數獨搏鬥——直到解答出來的那一刻，所獲得的成就感，真是筆墨所難以形容。

在學會電腦程式語言後，我開始構思如何設計一個簡捷且效能良好的小程序以破解數獨這遊戲，因為每次玩數獨遊戲，都是用鉛筆、橡皮擦不斷地歷經嘗試、失敗的過程，最後才能成功解題；因而開始思索這種「不斷地嘗試」的特性，如果交給電腦處理，一定能更為有效、快速的加以解決。此外，如果能自行設計出一個解題程式，當自己沒有空閒去解答數獨題目時，就可以交由我所設計的程式來進行破解，如此，也能與自己用手算推導出來的答案，擁有相同的成就感。

### 二、研究步驟與目的

提出一個可行的數獨解題演算法，藉由 **Visual Basic** 予以實作，再利用網路與報章上的數獨題目，進行程式正確性之驗證，之後試著將此一程式結合日常生活中可能遇到之問題，並實際應用此一程式加以運算，最終提出可行的解決方案。

## 貳●正文

### 一、數獨介紹

數獨，是一種填字遊戲，因為它名為「數」獨，許多人便會將它和數學聯想在一起，然而，玩數獨並不需要太好的數學功力，只要懂得寫出 0~9，再加上不屈不撓的毅力，就可以輕鬆的上手，算是個老少咸宜的小遊戲。無論何時何地，只要畫出一張 9X9 的表格，就可以感受數獨的獨特魅力！

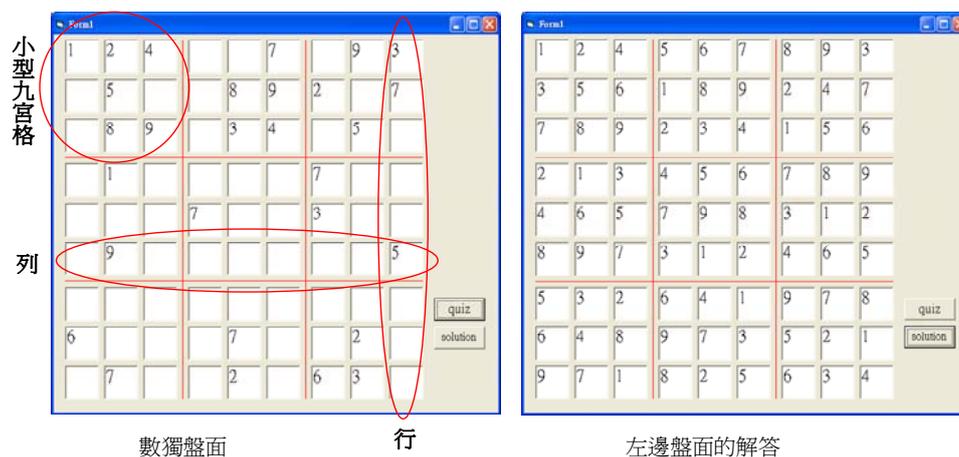
### 二、數獨的起源

數獨的祖先並不是魔術方陣（**magic square**，填滿整數的方陣，其中各行、各列、對角線的總和都必須一樣），這也許與一般人的認知或有不同。其實除了擁有數字和方陣這共同性外，數獨幾乎和魔術方陣沒有瓜葛[1]。「數獨」最早在 18 世紀後期由瑞士數學家 **Leonhard Euler** 所發明，命名為「拉丁方塊」；1984 年時，日本 **Nikoli** 雜誌，正式將此遊戲更名為「數獨」並開始連載[2]，從此數

獨就如燎原野火，燒向全球。

### 三、數獨的規則

數獨的遊戲規則是在一個 9 格寬 x 9 格高的盤面上，填入數字 1~9 使每行、每列及 3 格寬 x 3 格高的小型九宮格，都不能出現相同的數字，如下圖所示。



圖一 數獨盤面說明

### 四、數獨與生活

如果在 yahoo!奇摩網站搜尋列上鍵入關鍵字「數獨」，竟然能找到九百一十多萬個網頁，這實在是個非常驚人的數量，也證實了此遊戲已經蔓延到全世界，並且深植人心了！。

有個案例是這樣的：台北市一名專偷健身中心的女賊，竊得一本《麻辣數獨》，未料這本書籍引起她的高度興趣，她隨即在休息室開始填答了起來！[6]，可見數獨對人類的影響實在不容小覷，連偷東西的慣犯都會在犯罪時不自覺得迷上了呢！甚至有人在馬克杯上玩數獨，如下圖。



圖二 畫上數獨题目的馬克杯 [8]

曾發明六子棋的交大資訊系教授吳毅成邀請全球民眾利用個人電腦等多餘

計算資源共同計算數獨難題[7]，吳教授指出，如果有 10 萬個民眾願意加入，原本單核心電腦計算需要長達 30 萬年的時間才能解出的題目，將可縮短到 3 個月完成，由此可見，破解這個謎題需要大家的共同付出、努力。現代的人們，開始使用與電腦有關的各種技術，來破解與數獨有關的謎題，如果有心，再加上眾人的團結合作，那麼，數獨之謎將被一一解開！這也是本篇論文想要達成的任務。

## 五、數獨的解題法

常見的數獨解題法歸納如下表，但礙於篇幅所限不便詳細說明，請參考引註資料[3]中的較為詳細的解說。

表一 數獨解題法歸納表

解題法名稱	英文名稱
直觀解題法	Intuitive problem-solving
顯式唯一法	Naked Single
顯式三數集法	Naked Triplet
顯式四數集法	Naked Quad
隱式唯一法	Hidden Single
隱式數對法	Hidden Pair
隱式三數集法	Hidden Triplet
隱式四數集法	Hidden Quad
區塊刪減法	Intersection Removal
矩形對角線法	X-wing
XY 形態匹配法	XY-wing
XYZ 形態匹配法	XYZ-wing
WXYZ 形態匹配法	WXYZ-wing
三鏈數刪減法	Swordfish

## 六、解題程式設計

### (一) 程式設計使用之方法

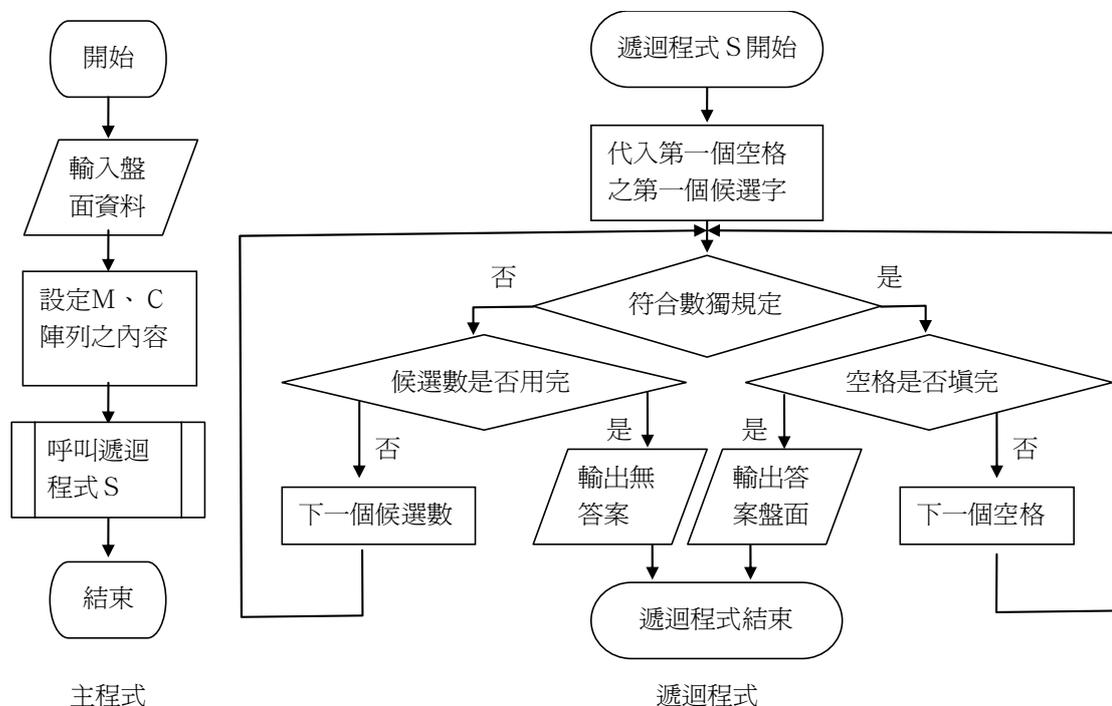
本論文的解題程式將運用遞迴的技巧，搭配候選數解題法，來進行設計。所謂遞迴是一個「可由自身所定義或呼叫的函數或程式」又可以分為以下 2 種形式：

- (1)直接遞迴(Direct recursion)：函數可以呼叫自身，如函數 A 內部又呼叫函數 A。
- (2)間接遞迴(Indirect recursion)：函數可以再度引用其呼叫函數，如函數 A 內部呼叫函數 B，而函數 B 內部呼叫函數 A，形成間接遞迴。

所謂候選數解題法，就是根據數獨的規則，在空格中填上可能是解答的數

字，我們稱此一數字為候選數，再根據候選數擇一做分析，正確則填上，反之則予以刪除。

### (二) 程式流程圖



圖三 解題程式流程圖

### (三) 程式演算法

Step 1: 設立兩個陣列，一個為  $81 \times 2$  的二維陣列，命名為M，負責儲存一開始的盤面資料，與是否已被選取，另一個為  $81 \times 9$  的二維陣列，命名為C，用於記載每個待填入空格中的候選數。

Step 2: 讀入原始盤面的數字，儲存於陣列M，並判讀陣列M之內容來進行設定陣列C的候選數。

Step 3: 將所有空格之所有候選數，依序進行判斷，如有一空格之所有候選數均已選過，則跳至 Step 6。

Step 4: 利用陣列C，依序進行候選數選擇，如符合數獨規定，填入陣列M，判斷所有空格是否均已填入數字，如是則結束跳至 Step 5，反之回到 Step 3 判斷下一個空格。如不符合數獨規定，將數字由陣列M內清除，回到 Step 3 選入下一個候選數。

Step 5: 將陣列M內之解答，輸出至盤面空格中。

Step 6: 輸出無解答之訊息。

#### (四) 實作方法

下列分述判斷數獨規定是否成立的三個核心程式碼，利用兩個 for 迴圈 p、q 變數執行 9X9 陣列共 81 格數字的判斷，x、y 代表輸入之行、列位置。

判斷是否符合同一列不重複的程式實作：

```

For p = 0 To 8
  For q = 0 To 8
    If m((y * 9 + p) Mod 9 + q * 9, 0) = tt Then Exit For
  Next
  If q = 9 Then
    f = 1
    X1 = ((y * 9 + p) Mod 9) \ 3
    Y1 = (y * 9 + p) \ 9 \ 3
    For a = 0 To 2
      For b = 0 To 2
        If M(X1 * 3 + Y1 * 27 + b * 9 + a, 0) = tt Then f = 0
      Next
    Next
    If f = 1 Then If M(y * 9 + p, 1) = 0 Then C(y * 9 + p, tt) = 0
  End If
Next

```

判斷是否符合同一行不重複的程式實作：

```

For p = 0 To 8
  For q = 0 To 8
    If m(((x + p * 9) \ 9) * 9 + q, 0) = tt Then Exit For
  Next
  If q = 9 Then
    f = 1
    X1 = ((x + p * 9) Mod 9) \ 3
    Y1 = (x + p * 9) \ 9 \ 3
    For a = 0 To 2
      For b = 0 To 2
        If M(X1 * 3 + Y1 * 27 + b * 9 + a, 0) = tt Then f = 0
      Next
    Next
    If f = 1 Then If M(x + p * 9, 1) = 0 Then C(x + p * 9, tt) = 0
  End If
Next

```

判斷是否符合同 3 格寬 x 3 格高的小型九宮格不重複的程式實作：

```

x = (i Mod 9) \ 3
y = i \ 9 \ 3
For a = 0 To 2
  For b = 0 To 2
    X2 = x * 3 + y * 27 + b * 9 + a
    For p = 0 To 8
      X1 = X2 Mod 9
      Y1 = X2 \ 9
      If M(Y1 * 9 + p, 0) = tt Then Exit For
      If M(X1 + p * 9, 0) = tt Then Exit For
    Next
    If p = 9 Then If M(X2, 1) = 0 Then C(X2, tt) = 0
  Next
Next

```

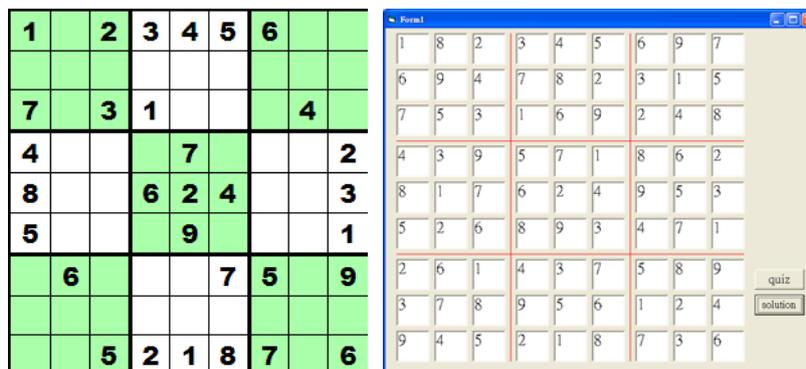
## 七、實驗結果

本論文所設計的解題程式，使用之實驗軟、硬體環境如下表所示：

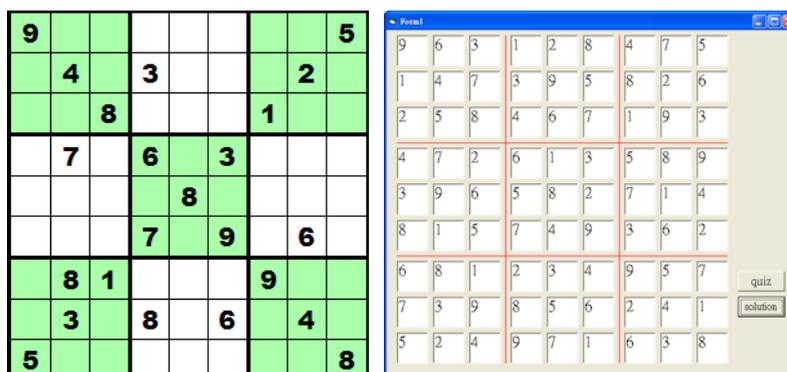
表二 本實驗之軟、硬體環境

中央處理器	Intel Core 2 Qual
記憶體	2GB DD2 RAM
作業系統	Windows XP sp3
設計程式語言	Visual Basic 6.0 sp6

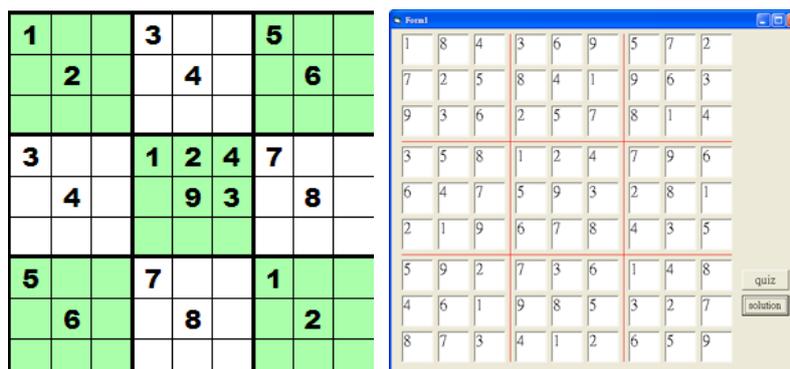
本論文利用網路上人氣極高的尤怪之家裡的經典數獨題目[5]，進行實驗，結果均能正確解出答案，但是如果遇到難度較高的題目，本程式的執行時間就會耗費相當久的運算時間，這應是演算法則未臻良好的因素，實驗結果如下圖。



圖四 尤怪之家的經典數獨與本程式解答之一



圖五 尤怪之家的經典數獨與本程式解答之二



圖六 尤怪之家的經典數獨與本程式解答之三

### 八、程式與現實生活應用的結合

本程式不僅能快速地破解數獨題目，還能運用到現實生活中，例如：某國立高職之綜合高中有九個班級，授課教師共有九位，每位教師因專長不同各自負責一門專業課程，並共設置九間不同的專業實習教室供授課使用，每個班級都得修完所有專業課程，要如何才能將教師們的授課時間表做出適當的分配呢？

我們先假設有甲~壬九個班級，1~9 九間專業教室，並且與我所設計的數獨解題程式之表單結合，經過運算後，可行的解決方案之一呈現如下圖：



圖七 九位教師排課表的解決方案

上圖中，盤面文字方塊裡的數字 1~9 即為九位老師綜合的課程表，就跟數獨的規則一樣，同一直排與橫排中，數字不得重複，而且，每一位教師可依個人的喜好，選定自己的課程表，再按下 solution，就如同改了個題目，程式依然能夠將其他教師的課程排入，實在非常方便！

#### 參●結論：

本論文試著提出一套可行的數獨解題演算法，並加以實作與驗證。雖然較複雜題目執行時，會需要大量的運算時間，但現階段而言，對大部分的坊間數獨題目，都有很好的解決效率。期待未來能深入學習、研究出更佳的演算法則，讓此一數獨解題程式發揮更好的運算效能。此外，經過這次的研究，也發現有些數獨題目並非只有單一解答，有些題目有著多個不同的解答。我將期許自己未來進行更深入的研究，讓此一程式，不僅能解出數獨題目，還能藉由各式規則的完備，自行產生出新的數獨盤面，以供所有的數獨愛好者一同研究、遊玩。

數獨不僅僅是一種遊戲，還能與生活結合，本論文也利用如上述綜合高中排課表的實例，來驗證此數獨程式的實用性，希望能對未來各式各樣的排班表，或者是各類的循環賽事，均能以此程式為主要核心，搭配不同的模式、參數設定，快速產生合乎規定的排列方式，如此一來，便不用人工勞心費神的安排了，只要將資料輸入介面中，定好規則與相關設定，就能提供使用者最適當的安排服務了。

肆●引註資料：

- [1] 林正峰(2005)，用「數獨」炒熱出版商機的推手，第 941 期，2005 年 12 月 5 日，商業周刊
- [2] 狄拉賀 撰文／翁秉仁 翻譯（2006），數獨樂，樂無窮，2006 年 7 月號，科學人
- [3] 愛數獨遊戲官方網站，2010 年 11 月，取自  
<http://www.acrystalman.com/cht/sudoku/index.htm>
- [4] Life and style，2010 年 11 月，取自  
<http://www.guardian.co.uk/lifeandstyle/2010/nov/03/sudoku-1709-medium>
- [5] 尤怪之家，2010 年 11 月，取自  
[http://163.19.31.11/cgi-bin/shes\\_sudoku/opensudo/opensudo.cgi?tth+1+2+6+16](http://163.19.31.11/cgi-bin/shes_sudoku/opensudo/opensudo.cgi?tth+1+2+6+16)
- [6] 突發中心(2009)，偷數獨玩入神 女賊被逮，蘋果日報，12 月 20 日，初版
- [7] 蕭介雲(2010)，解決數獨懸案 超大電腦計算邀你參加，台灣醒報，10 月 27 日，初版
- [8] 博客來創意生活館，2010 年 11 月，取自  
[http://www.books.com.tw/exep/prod/newprod\\_file.php?item=N000112853](http://www.books.com.tw/exep/prod/newprod_file.php?item=N000112853)
- [9] 黃世陽.吳明哲.何嘉益.張志成.吳志忠.曹祖聖 編著(1998)，Visual Basic 6.0 中文版學習範本，臺北市：松崗出版社