

投稿類別：資訊類

篇名：

SUDOKU!-解數獨

作者：

李冠廷。國立羅東高中。高一五班

指導老師：

邱柏翰老師

SUDOKU!-解數獨

壹●前言：

一、摘要：

數獨（Sudoku）是一種邏輯性的數字填充遊戲，規則為遊戲設計者會提供一部份的數字作為提示，玩家則須依提示以 1 至 9 所有數字填進每一格，而每行、每列和每個宮（即 3x3 的大格）均不得重複。

數獨廣受大眾歡迎，不論老幼，所有人都能輕鬆上手，但是，遇到稍具難度的題目，卻總令人煞費苦心，因此，製作出這個程式來更快的解出數獨。

二、研究動機：

在各式各樣的益智遊戲中，最廣為人知的莫過於是數獨了，數獨不只具有趣味性，也能夠鍛練個人的邏輯思考能力。但是，如果想要解開稍具難度的數獨題目，卻往往需要耗費大量腦力，也不一定解得出來，因此，想要將解數獨的過程模組化，製作成一個程式，利用電腦所具備的快速運算能力，來解開數獨。

貳●正文：

一、研究目的：

- (一) 尋找適當的演算法使數獨的解題過程模組化。
- (二) 尋找合適的程式語言進行該演算法，用於解題、驗證結果。

二、研究設備及器材：

| 硬體 | 軟體 |
|----|---------|
| 電腦 | Dev-C++ |

三、研究過程：

(一)解題過程

SUDOKU!-解數獨

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | | | 7 | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | | 6 | | | | 3 |
| 4 | | | 8 | | 3 | | | 1 |
| 7 | | | | 2 | | | | 6 |
| | 6 | | | | | 2 | 8 | |
| | | | 4 | 1 | 9 | | | 5 |
| | | | | 8 | | | 7 | 9 |

圖一：數獨題目

1.我原先的構想是以「尋找唯一可行解」為主，但當我寫好初步程式，並進行測試（以圖一的題目測試）時，卻遇到了挫折（如圖二）。

```
532678490
674195832
198342760
825964003
409853021
713020956
967530284
380419605
251086379
請按任意鍵繼續 . . .
```

圖二：測試結果(1) (0 代表空格)

2.經過我和老師的討論後，發現問題是出在「尋找唯一可行解」這個動作，因為並非所有題目都能夠找到唯一解，導致程式執行上會發生錯誤，在重新檢視題目及解題流程後，我決定將程式重新調整，

從「尋找唯一可行解」改成「尋找有解的可行解」，最後想出的解題步驟為：

- (1)尋找此題目的所有空格
- (2)各空格各自尋找個別之可行解
- (3)如可行解數目僅一個，則直接填入
- (4)如可行解數目超過一個，則先不填入，並將所有可能解以「題目」方式提供給子函數作判斷
- (5)如判斷結果為「有解」，再回傳給主函數進行填入

3.最後以圖一的題目進行測試，結果如圖三所示，證明此程式可行，能夠解開此數獨題目。

SUDOKU!-解數獨

```
    }  
}  
void solve(int s[][9],int m,int n)-----主函數  
{  
int k[10]={1,1,1,1,1,1,1,1,1,1};  
int f=0;  
int a,b,c,d,e,g,h;  
    for(a=0;a<9;a++)-----開始解題  
    {  
        for(c=((m/3)*3);c<(((m/3)*3)+3);c++)  
        {  
            for(d=((n/3)*3);d<(((n/3)*3)+3);d++)  
            {  
                if(s[a][n]!=0)-----橫向檢查  
                {  
                    k[s[a][n]]=0;  
                }  
                if(s[m][a]!=0)-----縱向檢查  
                {  
                    k[s[m][a]]=0;  
                }  
                if(s[c][d]!=0)-----宮(3X3 大格)內檢查  
                {  
                    k[s[c][d]]=0;  
                }  
            }  
        }  
    }  
    for(e=1;e<10;e++)  
    {  
        f=f+k[e];  
        if(f==1&& k[e]==1)-----唯一解，直接輸入  
        {  
            s[m][n]=e;  
        }  
        if(f!=1)
```

SUDOKU!-解數獨

```
{
    s[m][n]=0;
}
}
for(e=1;e<10;e++)
{
    f=f+k[e];
    if(f==2&& k[e]==1)-----非唯一解，呼叫子函數
    {
        if(test(s,m,n,e)!=1)-----如有解，再解開
        {
            s[m][n]=e;
        }
    }
}
}
int main(void)-----主程式
{
    int s[9][9];
    int a,b,c;
    for(a=0;a<9;a++)
    {
        for(b=0;b<9;b++)
        {
            cin>>s[a][b];-----輸入題目
        }
    }
    for(c=0;c<9;c++)
    {
        for(a=0;a<9;a++)
        {
            for(b=0;b<9;b++)
            {
                if(s[a][b]==0)-----尋找空格
                {
                    solve(s,a,b);-----呼叫主函數
```

SUDOKU!-解數獨

```
    }  
  }  
}  
}  
for(a=0;a<9;a++)  
{  
  for(b=0;b<9;b++)  
  {  
    cout<<s[a][b];-----輸出答案  
  }  
  cout<<"\n";  
}  
system("pause");  
return 0;  
}
```

參●結論：

目前此程式僅能針對「可能解數目不超過二」的題目進行破解，如可能解數目大於二時，就會產生錯誤，希望之後能再加以改進。

除此之外，在研究的過程中，我也發現坊間各種數獨題目，其提示數的數目十分多樣，對可能解的數目似乎也有影響，希望之後能研究「數獨的最少提示數」。而在製作此程式的過程中，也讓我瞭解到我的程式技巧還有很大的進步空間，希望之後能夠繼續精進。

肆●引註資料：

註一、洪維恩（2008）。C++ 教學手冊第二版。臺北市：博碩文化。

註二、數獨-維基百科。<http://zh.wikipedia.org/wiki/%E6%95%B8%E7%8D%A8>

註三、C 語言學習加油站。http://120.101.70.10/c_wiki/doku.php

註四、數獨樂園。<http://oddest.nc.hcc.edu.tw/sumain.htm>

註五、圖一來源：

http://blog.roodo.com/the_knowledge_walker/archives/1948582.html