

中華民國第四十六屆中小學科學展覽會
作品說明書

國中組 生活與應用科學科

030814

數獨酷-程式設計

學校名稱： 私立南山高中(附設國中)

作者： 國二 張胤翔 國二 黃奕翔 國二 簡捷 國二 高偉倫	指導老師： 辛明聰
--	--------------

關鍵詞：數獨、拉丁方陣、程式設計

壹、研究摘要

藉由檢視市面上流行的數獨書籍，統計分析數獨的組成結構及其相關模式。研究探討數獨題目的數字出現個數、分布情形及其對難易程度的影響，並且引用拉丁方陣的結構特性來進行數獨規則的研究，依據所研究規則運用於數獨程式的開發，並由電腦迅速產生數獨題目，這些電腦產生的數獨題目所具有的特性，是有別於市面上流行人工設計數獨問題的。經由這個研究，將數獨程式設計的原始碼公開，以提供進一步探究更深入數獨問題的基礎。



貳、研究動機

最近，學校裡興起一股數獨的風潮，許多老師跟同學課餘時間都在思考、求解數獨問題。雖然，報紙及數獨書本能提供許多的數獨問題，但是，購買數獨書籍或報紙，長期而言，對國中生也會造成經濟負擔，且題目有限。恰巧，上電腦課時，電腦老師提到數獨題目可以由電腦程式隨機產生，不限題數且不需花費，對於喜好求解數獨問題的我們，若能自行開發設計數獨程式，是一件新奇有趣的挑戰，於是展開了這趟探索之旅。

參、研究目的

想要分析、探求數獨問題的結構，並且能將分析的結果應用在程式設計上以解決數獨問題。在研究數獨的過程中，網路知識只提供對數獨問題的基本認識，對於數字出現的頻率、個數分布情形、難易程度的界定等，均無法提供相關知識。雖然網路也有一些數獨程式，不過均未將原始的程式碼公開，無法了解作者如何利用電腦來解決數獨問題。因此，藉由這次數獨題目的研究，分析探討如何產生滿足數獨條件的題目，並由電腦立即提供參考解答。透過 Flash 電腦軟體進行數獨程式的研究開發，並將研究成果及原始程式碼予以公開、發表，以做為更複雜數獨模組或更深入數獨問題研究的參考基礎。

肆、研究設備及器材

- 一、筆記型電腦兩台
- 二、Flash 程式
- 三、Gsp 程式
- 四、Microsoft PowerPoint
- 五、Microsoft Word
- 六、數獨(格林出版社)一至六冊

伍、研究過程及方法

在研究數獨的過程中，大部分都是由團隊成員共同來進行問題探討的，從題目的觀察、求解，資料的統計、分析，到模式的研究及探討等，而網路上所能提供的數獨知識，對這個研究的幫助是非常有限的。我們採用目前流行的格林版數獨本(整套含第一集到第六集)進行觀察紀錄，再藉由試算軟體進行相關資料的統計分析，最後予以彙整，並嘗試找出數獨題目的結構與規則。

一、數獨的規則

給定一個九列九行的九階(9*9)方陣，其中每一格裡都要填入 1 或 2 或 3 或 4 或 5 或 6 或 7 或 8 或 9 的整數，且要滿足下列條件：

- (一)每一行及每一列分別填入數字 1~9
- (二)同一行不能有相同的數字
- (三)同一列不能有相同的數字
- (四)同一個九宮格內不能有相同的數字 (註):參考九宮格對照表
- (五)要把 81 個格子完全填滿

二、數獨的九宮格圖

九宮格對照表

上左 九宮格	上中 九宮格	上右 九宮格
中左 九宮格	中央 九宮格	中右 九宮格
下左 九宮格	下中 九宮格	下右 九宮格

三、數獨的難易度

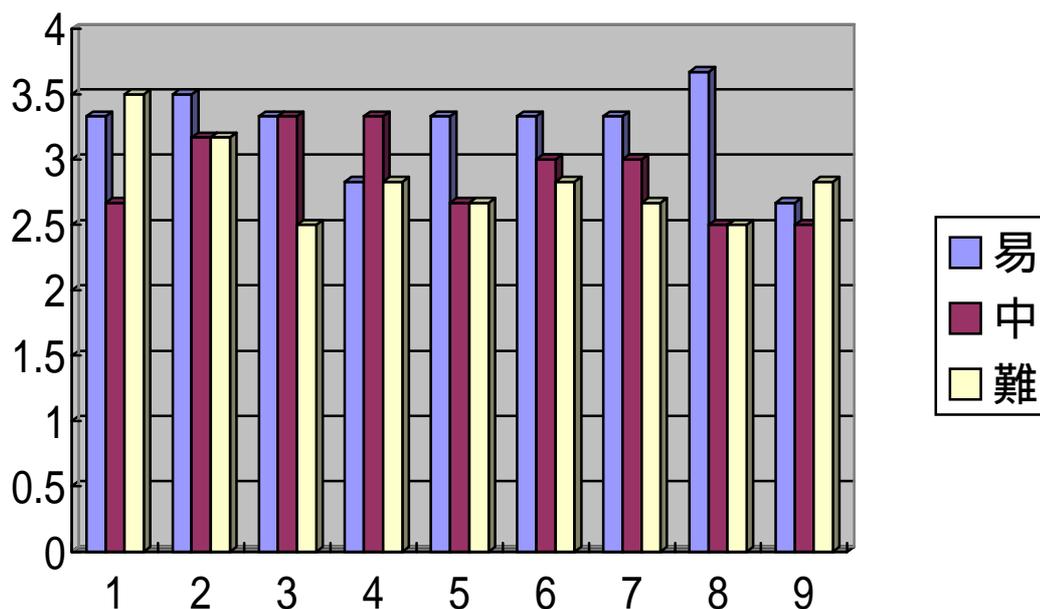
將市面上頗為流行的格林版數獨集(1~6集)，進行數字出現率與數獨難易度的統計分析，藉由統計的結果，觀察其難易度與數字出現率的關係。研究觀察發現，數獨的難易度跟給予的數字多寡會在一定的範圍內有關聯性，僅少數極端性範例與數字多寡較無相關(極端性範例通常為人工設計的特例)，而是與其分布模式有直接的關聯。各個數字的出現率在數獨的各種問題上較無關聯。

根據數獨(格林版)統計分析表如下：

數獨格林板第一集出現數字統計表

	1	2	3	4	5	6	7	8	9
易	3.33	3.50	3.33	2.83	3.33	3.33	3.33	3.67	2.67
中	2.67	3.17	3.33	3.33	2.67	3.00	3.00	2.50	2.50
難	3.50	3.17	2.50	2.83	2.67	2.83	2.67	2.50	2.83

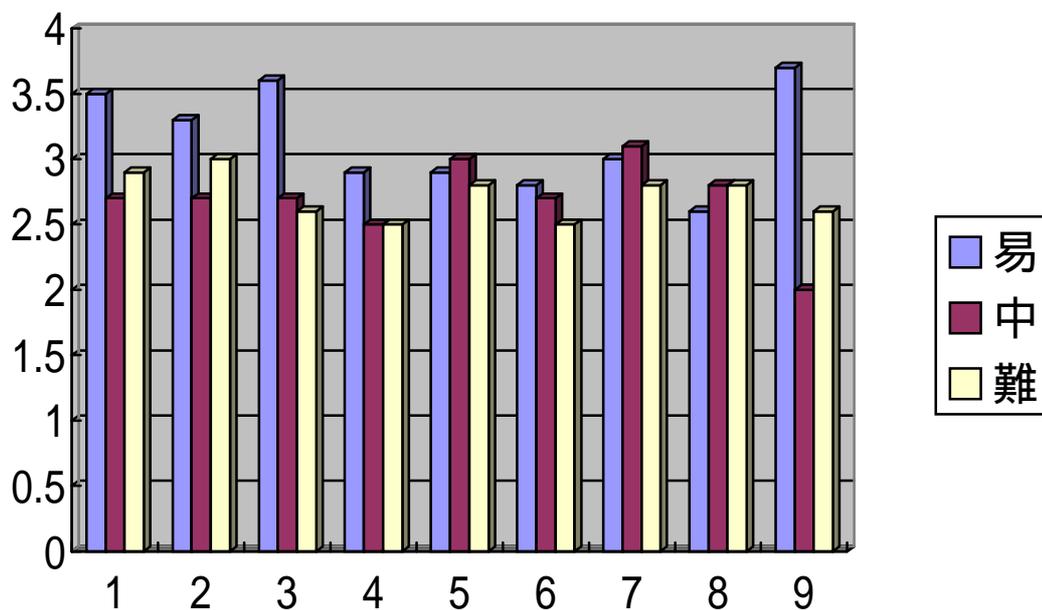
數獨格林板第一集出現數字統計圖



數獨格林版第二集出現數字統計表

	1	2	3	4	5	6	7	8	9
易	3.5	3.3	3.6	2.9	2.9	2.8	3	2.6	3.7
中	2.7	2.7	2.7	2.5	3	2.7	3.1	2.8	2
難	2.9	3	2.6	2.5	2.8	2.5	2.8	2.8	2.6

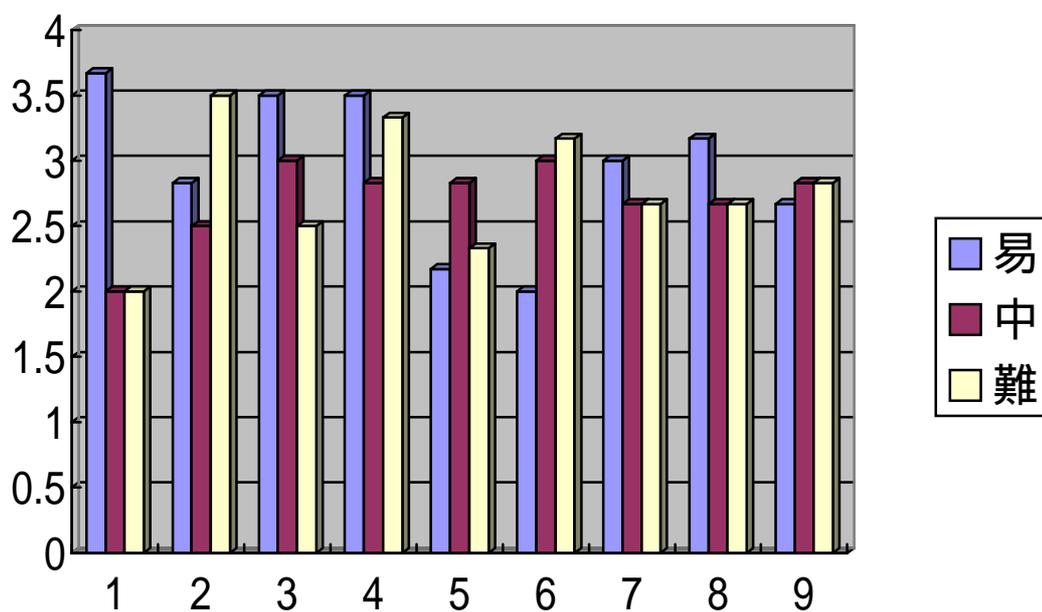
數獨格林版第二集出現數字統計圖



數獨格林版第三集出現數字統計表

	1	2	3	4	5	6	7	8	9
易	3.67	2.83	3.50	3.50	2.17	2.00	3.00	3.17	2.67
中	2.00	2.50	3.00	2.83	2.83	3.00	2.67	2.67	2.83
難	2.00	3.50	2.50	3.33	2.33	3.17	2.67	2.67	2.83

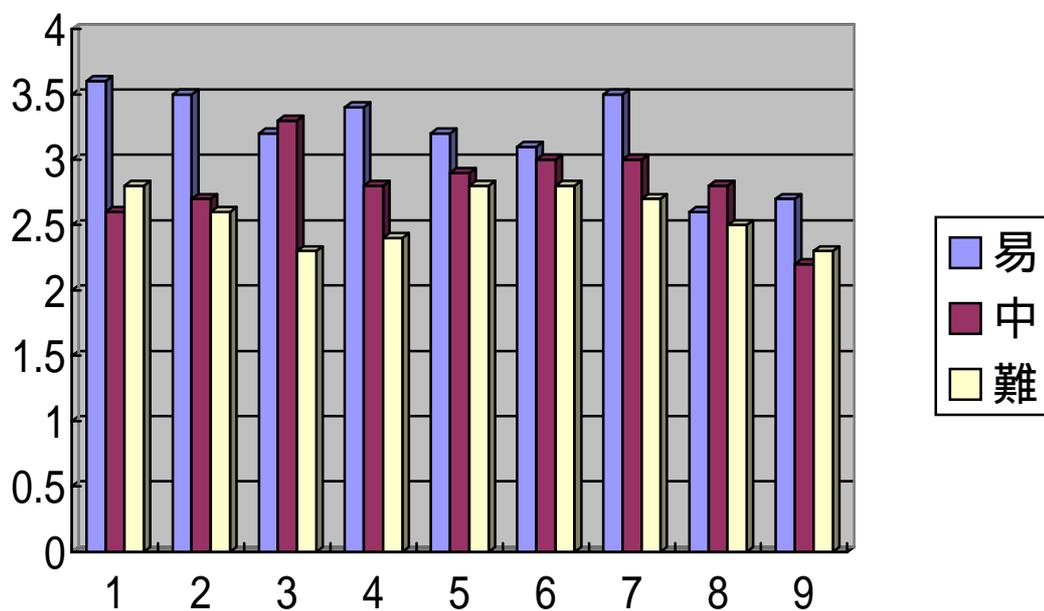
數獨格林版第三集出現數字統計圖



數獨格林版第四集出現數字統計表

	1	2	3	4	5	6	7	8	9
易	3.6	3.5	3.2	3.4	3.2	3.1	3.5	2.6	2.7
中	2.6	2.7	3.3	2.8	2.9	3	3	2.8	2.2
難	2.8	2.6	2.3	2.4	2.8	2.8	2.7	2.5	2.3

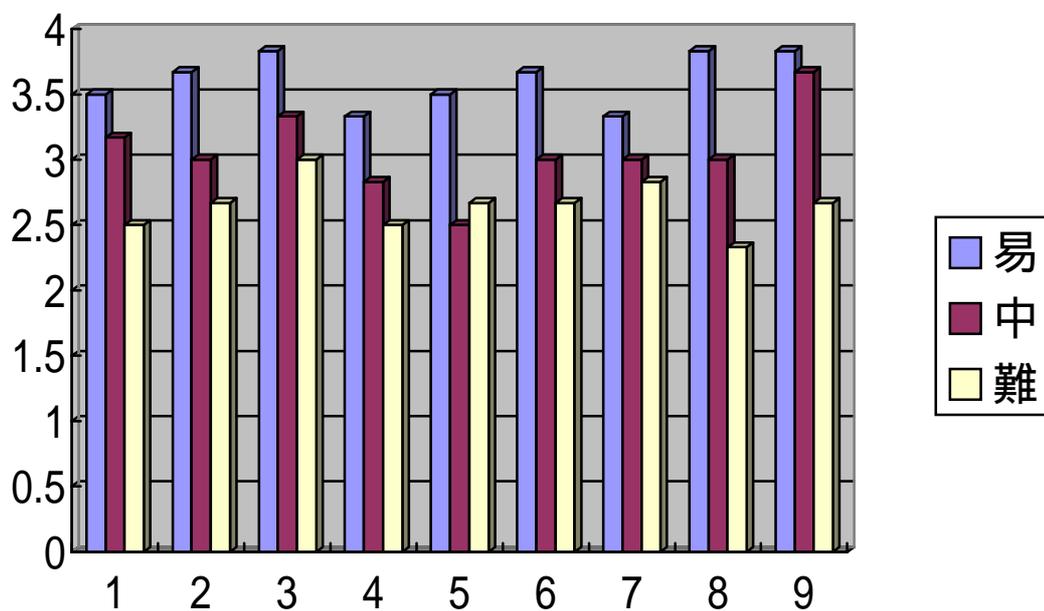
數獨格林版第四集出現數字統計圖



數獨格林版第五集出現數字統計表

	1	2	3	4	5	6	7	8	9
易	3.50	3.67	3.83	3.33	3.50	3.67	3.33	3.83	3.83
中	3.17	3.00	3.33	2.83	2.50	3.00	3.00	3.00	3.67
難	2.50	2.67	3.00	2.50	2.67	2.67	2.83	2.33	2.67

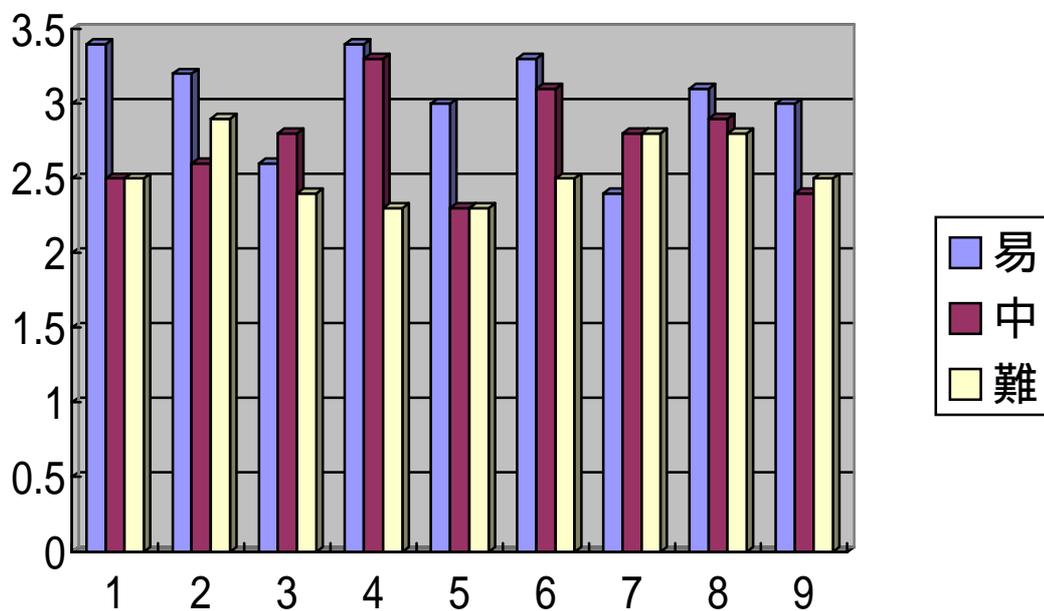
數獨格林版第五集出現數字統計圖



數獨格林版第六集出現數字統計表

	1	2	3	4	5	6	7	8	9
易	3.4	3.2	2.6	3.4	3	3.3	2.4	3.1	3
中	2.5	2.6	2.8	3.3	2.3	3.1	2.8	2.9	2.4
難	2.5	2.9	2.4	2.3	2.3	2.5	2.8	2.8	2.5

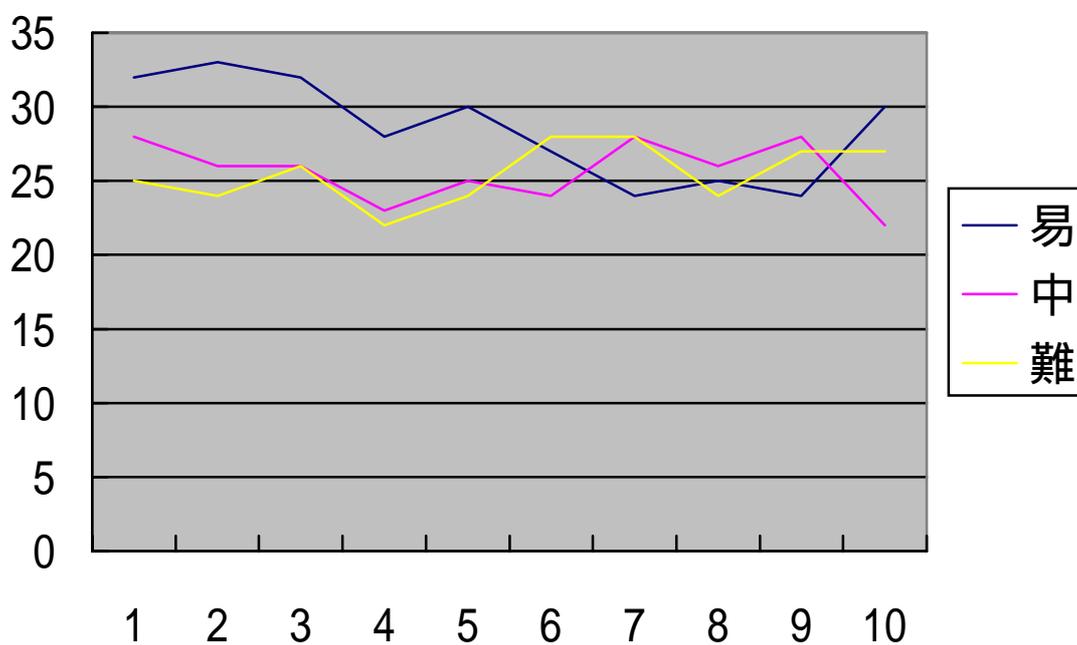
數獨格林版第六集出現數字統計圖



數獨格林版第一集每則個數統計表

	1	2	3	4	5	6	7	8	9	10
易	32	33	32	28	30	27	24	25	24	30
中	28	26	26	23	25	24	28	26	28	22
難	25	24	26	22	24	28	28	24	27	27

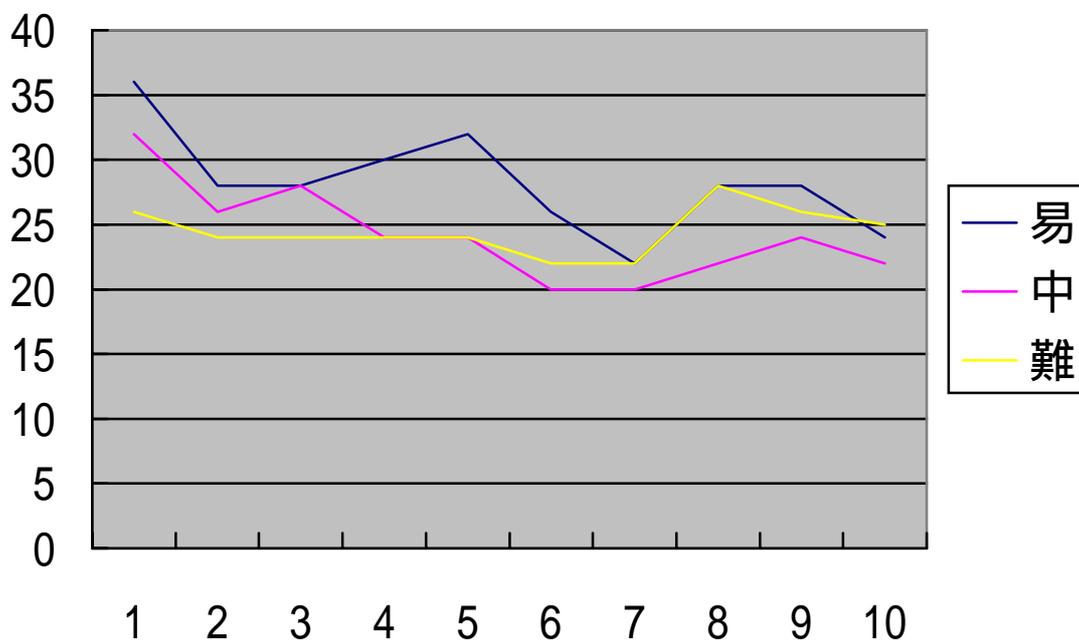
數獨格林版第一集每則個數統計圖



數獨格林版第二集每則個數統計表

	1	2	3	4	5	6	7	8	9	10
易	36	28	28	30	32	26	22	28	28	24
中	32	26	28	24	24	20	20	22	24	22
難	26	24	24	24	24	22	22	28	26	25

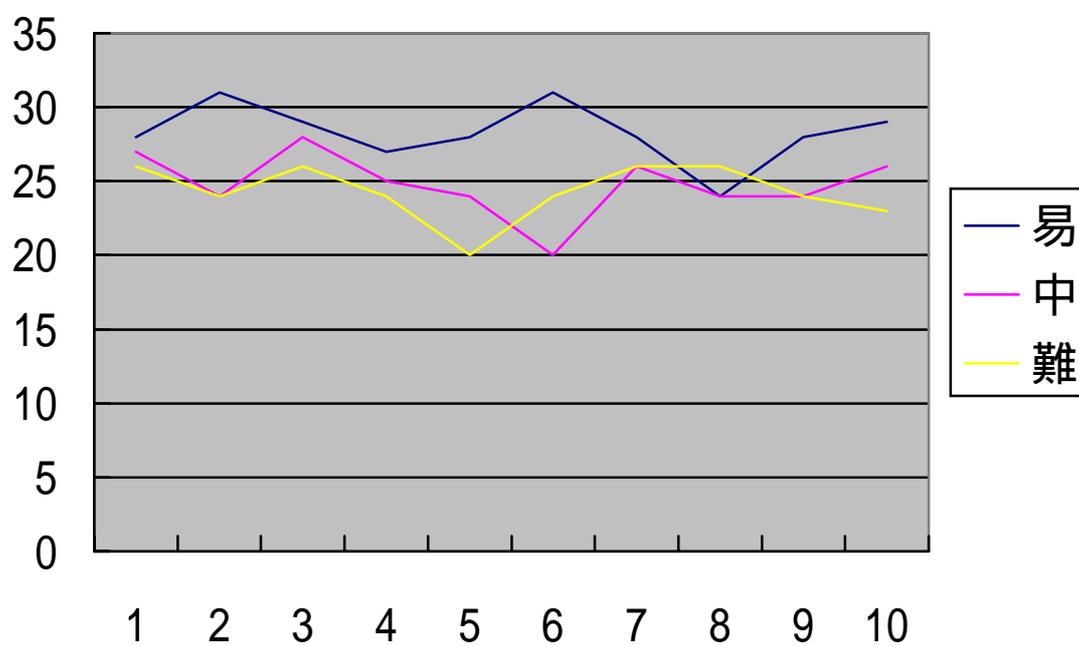
數獨格林版第二集每則個數統計圖



數獨格林版第三集每則個數統計表

	1	2	3	4	5	6	7	8	9	10
易	28	31	29	27	28	31	28	24	28	29
中	27	24	28	25	24	20	26	24	24	26
難	26	24	26	24	20	24	26	26	24	23

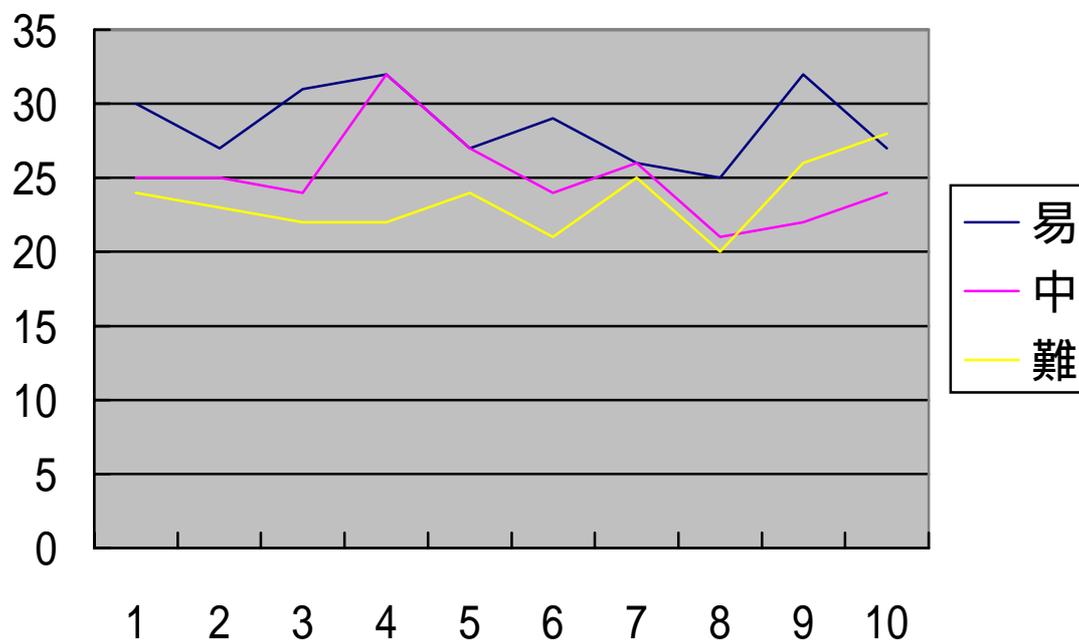
數獨格林版第三集每則個數統計圖



數獨格林版第四集每則個數統計表

	1	2	3	4	5	6	7	8	9	10
易	30	27	31	32	27	29	26	25	32	27
中	25	25	24	32	27	24	26	21	22	24
難	24	23	22	22	24	21	25	20	26	28

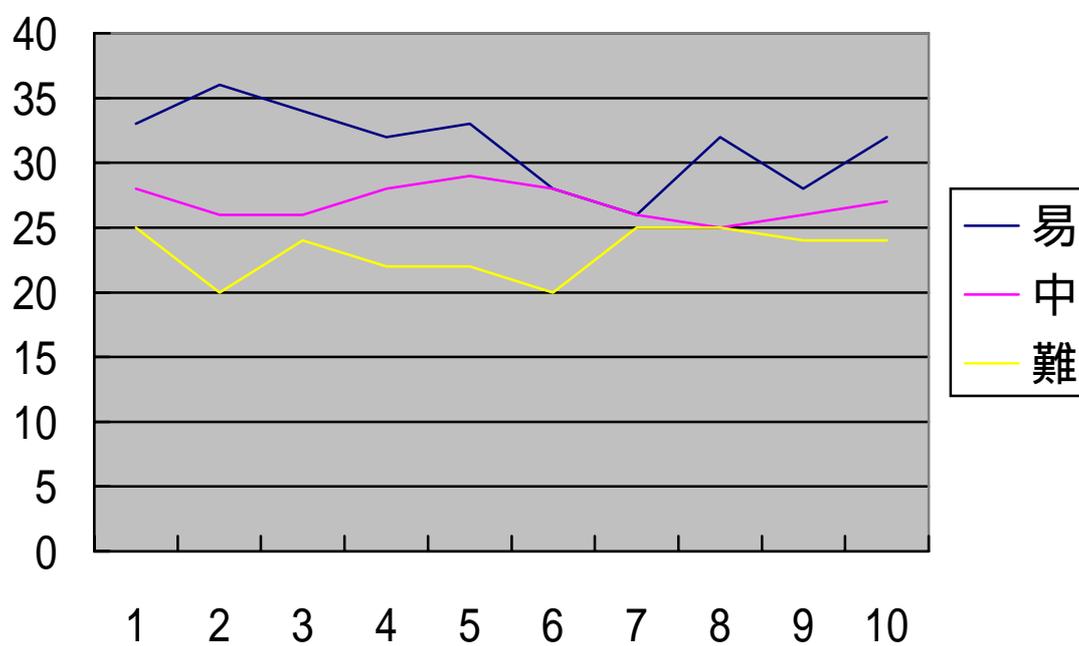
數獨格林版第四集每則個數統計圖



數獨格林版第五集每則個數統計表

	1	2	3	4	5	6	7	8	9	10
易	33	36	34	32	33	28	26	32	28	32
中	28	26	26	28	29	28	26	25	26	27
難	25	20	24	22	22	20	25	25	24	24

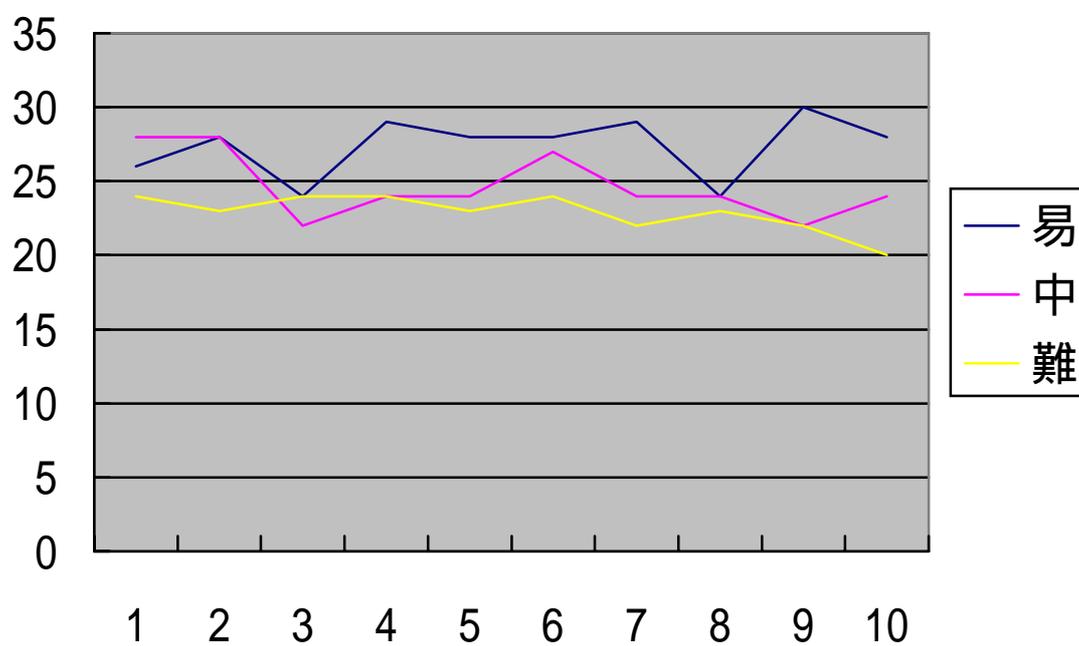
數獨格林版第五集每則個數統計圖



數獨格林版第六集每則個數統計表

	1	2	3	4	5	6	7	8	9	10
易	26	28	24	29	28	28	29	24	30	28
中	28	28	22	24	24	27	24	24	22	24
難	24	23	24	24	23	24	22	23	22	20

數獨格林版第六集每則個數統計圖



四、以拉丁方陣的結構來分析設計數獨題目

一個九階(9*9)的拉丁方陣，以數字 1~9 來取代原來的九個拉丁符號，並將其填入中央九宮格中，可以得到 36 萬 2880 種方法，再參照數獨的規則，依次將剩下的八個小九宮格填入適當的數字，即可得到一組 9*9 的數獨解答。(參考文獻：科學月刊 37 卷 3 期 214 頁)

根據上述規則，著手加以研究開發出一套能迅速產生數獨題目，且可以立即提供參考解答的數獨程式系統，透過 Flash 軟體進行本系統的原始程式碼的製作、撰寫。說明如下圖：

			A					
			B					
			C					
D	E	F	9	1	8	G	H	I
			7	2	4			
			5	6	3			
			J					
			K					
			L					

(一)方式 1

首先，先在 D,E,F 任意填入 2~7，選出 3 個數填入，(其中 9.1.8 不可填入)，填好後，剩餘的數字便填入 G,H,I，這一行便完成了。接著，再將 A,B,C 任意填入 1.2.3.4.6.8 其中三個數字(9.7.5 不可填入)，剩餘的數字也填入 J,K,L，這一系列便完成了。

依據這個規則，中央九宮格周圍的四個九宮格都可以被填入適當的數字，至於角落的四個九宮格，只要符合一開始所講的不重複原則就可以了。因為中央九宮格可以任意填入 1~9 九個相異數字，所以，就會產生各式各樣的數獨題目。

(二)方式 2

結合數學的平移、旋轉及鏡射等性質，來產生不重複且符合數獨規則的數字。可由方式 2 的規則產生第一個小宮格後(不必為中央九宮格)，將其他各個小宮格也以相同的規則置入其他不重複的數字，即可產生一符合數獨規則的題目。

文字說明如下：

1
8 9
3 2 4
7 6
5

以 2 為中心點，上方為 1、下方為 5、右方為 4、左方為 3，右上左上分別是 9 和 8，右下左下分別是 6 和 7，採用本方法是先於 1~9 中選定某一個固定的數字，並將此數依據數獨的規則分別任意填入九個小宮格中，重複應用左手邊的數字排定法則，直到超出宮格邊緣時，再將該數字填回該小宮對應的的格子中，這個規則不僅可以用來設計九階(9*9)型式的數獨問題

題

，甚至應用於重疊數獨型、巨無霸數獨等特殊型式的數獨問題上，也是可以的。

(三)方式 2 的說例

步驟 1：定義出數字 2(中心點)的位置

步驟 2：按照順序填入數字

若遇無空格可填如左上九宮格中心點 2 左方無格子可填，可將數字 3 移至空白處再填入左上宮格之相對位置(詳細資料請見記錄本)

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

(四)方式 2 的拓展及變化

此類型題目較為簡單不過可掉換單行、單列之位置以求變化(必須是對應格),也發現單行位置變換後不能再以單列形式變化,這裡還需要多花時間研究,或許斜對稱也可以產生對換,不過我們尚未發現其他把題目複雜化的方法,但此方法可應用在許多數讀題目(出題方面)拼圖數獨、彩色數獨還要我們在多製造幾組來比較。

(變換位置圖例請見下圖)

第 1 列	第 2 列	第 3 列	第 4 列	第 5 列	第 6 列	第 7 列	第 8 列	第 9 列
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

1.列調換法

圖中第 1、4、7 列和第 2、5、8 列與第 3、6、9 列為對應列
可對換位置(左右平移)

第 1 行
第 2 行
第 3 行
第 4 行
第 5 行
第 6 行
第 7 行
第 8 行
第 9 行

2.行調換法

圖中第 1、4、7 行和第 2、5、8 行與第 3、6、9 行為對應行
可對換位置(上下平移)

陸、 研究結果

經過上述研究發現，簡單型數獨問題，給予的數字個數約在 29 33 之間，中等型數獨問題，給予的數字個數約在 24 28 之間，困難型數獨問題，給予的數字個數約在 20 26 之間。當所給的數獨問題其數字個數落在中等及困難型的範圍之間時，其難易度是受到數字分布模式影響的，大多數電腦程式產生的數獨題目較符合統計分析的結果，只有少數極端的數獨問題不能由所給予的數字個數範圍來判定其難易度，這些特例往往跟出題的結構有關(如明顯的對稱、規則及平移等)，跟給予的數字多寡沒有明顯的關聯。

由統計發現，簡單型和中等型的數獨題目，數字分佈較為平均，每宮格約出現 3 4 個數字。難度高的題目數字的分布有極端化的傾向，也就是說密集的宮格，大多含有 3~4 個以上的數字，但是不密集的宮格，卻連 1 個數字都沒有出現，甚至出現兩個宮格內都沒有數字的情況。

此外，觀察數字 1 9，在各類數獨問題中的出現率，數字 1 9 的平均出現率差別不大，既使缺漏一個數字，對難易度也不會有太大的影響，經過統計分析發現數字出現的個數與數獨難易程度並無明顯的相關性。數字較多會傾向容易，稍少卻沒有明顯傾向困難(中等型或困難型都有可能)，在本研究中已經將這部分的問題予以處理。

柒、結論

一開始，我們猜測數獨的個數對難易度有較大的影響，經由統計結果發現，數字的個數只會在特定的範圍內對數獨難易度有所關聯，特別是程式設計下的數獨題目大致符合統計的結果。

數獨的難易度除了受到數字個數影響以外，數獨的結構與模式也會對難易度產生影響，於是引進科學月刊上探討數獨問題的論文，其中是以拉丁方陣的模式來分析數獨結構，我們便採用這種數獨模式來撰寫程式，不但獲得突破性的發展，並且根據數獨分布的情形，初步界定問題的難易度，並將數獨程式開發完成。可以迅速創造出新的數獨題目，且能快速地得到數獨的參考解答。

捌、未來展望

雖然我們的研究只是數獨的一種基本模式，且已獲得初步的成果，我們將研究所得及程式碼公開，就是希望這個程式能做為一個基礎，提供給想要進一步研究、了解數獨的人參考。除了本篇探討的拉丁方陣模式外，還有許多可延伸的種類值得探討，也希望未來能以電腦程式來呈現。以下我們思考了三種不同的模式，希望能吸引更多對數獨有興趣的人一同鑽研。

重疊數獨

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

巨無霸數獨

	6							10	
8		5		12	1			7	11
	4			5					8
			3	1	7	2	8	12	
		2	8					7	9
	5		12		3	6		1	2
	7		10		6	5		2	1
		9	6					11	3
			4	3	2	7	10	5	
	8			11			5		3
1		7			8	10			4
	3								12

拼圖數獨

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

玖、 程式碼及畫面 簡易型範例



中等型範例



困難型範例



拾、主要程式碼

```
on(press){
    for(i=1 ; i<=81 ; i=i+1){
        colr=255*256*256+255*256+255;
        eval("su"+i).backgroundColor=colr
        eval("su"+i).type = "dynamic";
        eval("su"+i).textColor=3382428;
    }
    kind=int(1+10*Math.random());
    if(kind==1){
        SD=new Array ();
        SDK=new Array ();
        KILL=new Array ();
        SOL=new Array ();
        for(i=1 ; i<=9 ; i=i+1){
            SD[i]=int(1+9*Math.random());
            for(j=1 ; j<i ; j=j+1){
                j="+j+" SD["+j+"]="+SD[j];
                m=SD[i];
                n=SD[j];
                if(m==n){
                    i=i-1;
                    break;
                }
            }
        }
        in31=SD[1];in32=SD[2];in33=SD[3];in40=SD[4];in41=SD[5];in42=SD[6];in49=SD[7];in50=SD[8];in51=SD[9];
        in46=SD[1];in47=SD[2];in48=SD[3];in28=SD[4];in29=SD[5];in30=SD[6];in37=SD[7];in38=SD[8];in39=SD[9];
        in43=SD[1];in44=SD[2];in45=SD[3];in52=SD[4];in53=SD[5];in54=SD[6];in34=SD[7];in35=SD[8];in36=SD[9];
        in6=SD[1]; in4=SD[2]; in5=SD[3] ;in15=SD[4];in13=SD[5];in14=SD[6];in24=SD[7];in22=SD[8];in23=SD[9];
        in59=SD[1];in60=SD[2];in58=SD[3];in68=SD[4];in69=SD[5];in67=SD[6];in77=SD[7];in78=SD[8];in76=SD[9];
        in11=SD[1];in10=SD[2];in19=SD[3];in2=SD[4] ;in21=SD[5];in20=SD[6];in3=SD[7]; in12=SD[8];in1=SD[9];
        in26=SD[1];in25=SD[2];in17=SD[3];in27=SD[4];in9=SD[5]; in8=SD[6]; in18=SD[7];in7=SD[8]; in16=SD[9];
        in75=SD[1];in66=SD[2];in74=SD[3];in57=SD[4];in73=SD[5];in55=SD[6];in65=SD[7];in64=SD[8];in56=SD[9];
        in72=SD[1];in81=SD[2];in70=SD[3];in80=SD[4];in61=SD[5];in79=SD[6];in62=SD[7];in63=SD[8];in71=SD[9];

        SOL[1]=in1; SOL[2]=in2; SOL[3]=in3; SOL[4]=in4; SOL[5]=in5; SOL[6]=in6; SOL[7]=in7; SOL[8]=in8; SOL[9]=in9;
        SOL[10]=in10;SOL[11]=in11;SOL[12]=in12;SOL[13]=in13;SOL[14]=in14;SOL[15]=in15;SOL[16]=in16;SOL[17]=in17;SOL[18]=in18;
        SOL[19]=in19;SOL[20]=in20;SOL[21]=in21;SOL[22]=in22;SOL[23]=in23;SOL[24]=in24;SOL[25]=in25;SOL[26]=in26;SOL[27]=in27;
```

```
SOL[28]=in28;SOL[29]=in29;SOL[30]=in30;SOL[31]=in31;SOL[32]=in32;SOL[33]=in33;SOL[34]=in34;SOL[35]=in35;SOL[36]=in36;
SOL[37]=in37;SOL[38]=in38;SOL[39]=in39;SOL[40]=in40;SOL[41]=in41;SOL[42]=in42;SOL[43]=in43;SOL[44]=in44;SOL[45]=in45;
SOL[46]=in46;SOL[47]=in47;SOL[48]=in48;SOL[49]=in49;SOL[50]=in50;SOL[51]=in51;SOL[52]=in52;SOL[53]=in53;SOL[54]=in54;
SOL[55]=in55;SOL[56]=in56;SOL[57]=in57;SOL[58]=in58;SOL[59]=in59;SOL[60]=in60;SOL[61]=in61;SOL[62]=in62;SOL[63]=in63;
SOL[64]=in64;SOL[65]=in65;SOL[66]=in66;SOL[67]=in67;SOL[68]=in68;SOL[69]=in69;SOL[70]=in70;SOL[71]=in71;SOL[72]=in72;
SOL[73]=in73;SOL[74]=in74;SOL[75]=in75;SOL[76]=in76;SOL[77]=in77;SOL[78]=in78;SOL[79]=in79;SOL[80]=in80;SOL[81]=in81;
```

```
if ( type==0 ) {      max_count=81-(29+int(1+4*Math.random()));}
else if ( type==1 ) { max_count=81-(24+int(1+4*Math.random()));}
else if ( type==2 ) { max_count=81-(19+int(1+4*Math.random()));}
    for(i=1 ; i<=max_count ; i=i+1){
        KILL[i]=int(1+81*Math.random());
        for(j=1 ; j<i ; j=j+1){
            m=KILL[i];
            n=KILL[j];
            if(m==n){ i=i-1;break; }
        }
    }
```

```
SDK[1]=in1; SDK[2]=in2; SDK[3]=in3; SDK[4]=in4; SDK[5]=in5; SDK[6]=in6; SDK[7]=in7; SDK[8]=in8; SDK[9]=in9;
SDK[10]=in10;SDK[11]=in11;SDK[12]=in12;SDK[13]=in13;SDK[14]=in14;SDK[15]=in15;SDK[16]=in16;SDK[17]=in17;SDK[18]=in18;
SDK[19]=in19;SDK[20]=in20;SDK[21]=in21;SDK[22]=in22;SDK[23]=in23;SDK[24]=in24;SDK[25]=in25;SDK[26]=in26;SDK[27]=in27;
SDK[28]=in28;SDK[29]=in29;SDK[30]=in30;SDK[31]=in31;SDK[32]=in32;SDK[33]=in33;SDK[34]=in34;SDK[35]=in35;SDK[36]=in36;
SDK[37]=in37;SDK[38]=in38;SDK[39]=in39;SDK[40]=in40;SDK[41]=in41;SDK[42]=in42;SDK[43]=in43;SDK[44]=in44;SDK[45]=in45;
SDK[46]=in46;SDK[47]=in47;SDK[48]=in48;SDK[49]=in49;SDK[50]=in50;SDK[51]=in51;SDK[52]=in52;SDK[53]=in53;SDK[54]=in54;
SDK[55]=in55;SDK[56]=in56;SDK[57]=in57;SDK[58]=in58;SDK[59]=in59;SDK[60]=in60;SDK[61]=in61;SDK[62]=in62;SDK[63]=in63;
SDK[64]=in64;SDK[65]=in65;SDK[66]=in66;SDK[67]=in67;SDK[68]=in68;SDK[69]=in69;SDK[70]=in70;SDK[71]=in71;SDK[72]=in72;
SDK[73]=in73;SDK[74]=in74;SDK[75]=in75;SDK[76]=in76;SDK[77]=in77;SDK[78]=in78;SDK[79]=in79;SDK[80]=in80;SDK[81]=in81;
```

```
for(i=1 ; i<=81 ; i=i+1){
```

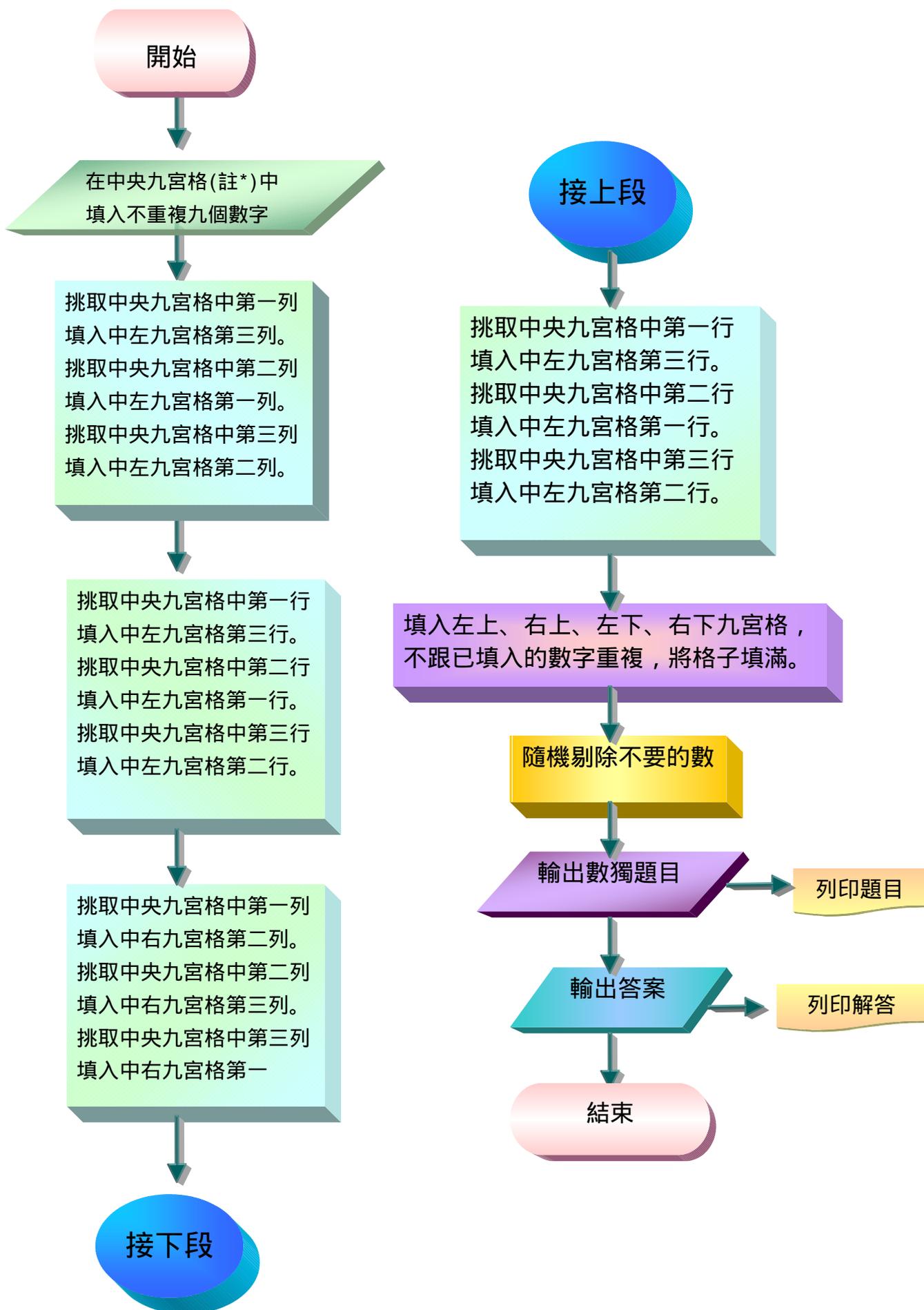
```
    SDK[KILL[i]]= " ";
```

```
}
```

```
in1=SDK[1]; in2=SDK[2]; in3=SDK[3]; in4=SDK[4]; in5=SDK[5]; in6=SDK[6]; in7=SDK[7]; in8=SDK[8]; in9=SDK[9];
in10=SDK[10];in11=SDK[11];in12=SDK[12];in13=SDK[13];in14=SDK[14];in15=SDK[15];in16=SDK[16];in17=SDK[17];in18=SDK[18];
in19=SDK[19];in20=SDK[20];in21=SDK[21];in22=SDK[22];in23=SDK[23];in24=SDK[24];in25=SDK[25];in26=SDK[26];in27=SDK[27];
in28=SDK[28];in29=SDK[29];in30=SDK[30];in31=SDK[31];in32=SDK[32];in33=SDK[33];in34=SDK[34];in35=SDK[35];in36=SDK[36];
in37=SDK[37];in38=SDK[38];in39=SDK[39];in40=SDK[40];in41=SDK[41];in42=SDK[42];in43=SDK[43];in44=SDK[44];in45=SDK[45];
in46=SDK[46];in47=SDK[47];in48=SDK[48];in49=SDK[49];in50=SDK[50];in51=SDK[51];in52=SDK[52];in53=SDK[53];in54=SDK[54];
in55=SDK[55];in56=SDK[56];in57=SDK[57];in58=SDK[58];in59=SDK[59];in60=SDK[60];in61=SDK[61];in62=SDK[62];in63=SDK[63];
in64=SDK[64];in65=SDK[65];in66=SDK[66];in67=SDK[67];in68=SDK[68];in69=SDK[69];in70=SDK[70];in71=SDK[71];in72=SDK[72];
in73=SDK[73];in74=SDK[74];in75=SDK[75];in76=SDK[76];in77=SDK[77];in78=SDK[78];in79=SDK[79];in80=SDK[80];in81=SDK[81];
```

```
}on
```

數獨程式製作流程圖



拾壹、參考文獻

- 一、科學月刊,第 37 卷,第 3 期,清華大學教授林克瀛
- 二、威震四方工作小組,數獨,格林版 1~6,格林文化有限公司,2005 年 12 月
- 三、PCREATION,解數獨,尖端出版 城邦文化有限公司,2005 年 9 月
- 四、旅行家工作室,國中電腦(二),台北市,碁峰出版社,2005 年 4 月初版
- 五、羅賓威爾森,如何解數獨,羅賓威爾森/著格林文化,2005 年 9 月 5 日
- 六、腦力潛能小組,玩數獨,尖端,2006 年 1 月 4 日
- 七、Nikoli/著,數獨 365,格林文化,2005 年 11 月 24 日
- 八、<http://www.sudoku.com/>
- 九、<http://tw.knowledge.yahoo.com/question/?qid=1105041406006>
- 十、<http://tw.knowledge.yahoo.com/question/?qid=1305092315885>
- 十一、<http://tw.knowledge.yahoo.com/question/?qid=1306020416816>
- 十二、<http://tw.knowledge.yahoo.com/question/?qid=1105053004053>
- 十三、<http://www.paps.kh.edu.tw/asp/math/9x9Number/Number.html>
- 十四、<http://infoserv.com.tw/phorum/read.php?f=1&i=55307&t=55061>

評 語

030814 數獨酷-程式設計

利用電腦製作程式解決數學問題創意性不錯，畫面設計優美。但對於問題難易度之量化定義不夠清楚。另可考慮不同型別之統計資料；若能設計解題方法則更好。