2025 年【科學探究競賽-這樣教我就懂】

□國中組　■普高組　□技高組　成果報告格式

題目名稱：手機螢幕 3×3 圖形鎖的所有情況數分析

一、摘要

1、分析 $1\times n$ 密碼的規律，總共有 $2^1(n-1)+2^2(n-2)+2^3(n-3)+\cdots+2^{n-1}\times1=2^n-2n-2$ 種。

2、分析 2×2 與 2×3 密碼，依序有 60、1378 種。

3、使用 Python 寫程式，計算 3×3 密碼，四點至九點的密碼總共有 389112 種，兩點至九點的密碼總共有 389488 種。

二、探究題目與動機

Android 手機螢幕鎖有一種方式是使用圖像密碼，由 3×3 的點組成的九宮格依照規定連線，我們好奇這種類型的密碼有多少種可能，利用【排列組合】、【程式設計】來計算所有可能性為本研究重點。

三、探究目的與假設

探索 $1\times n$ 、 $2\times n$ 圖形密碼可能性，試圖找出 3×3 圖形密碼可能性並利用程式驗算。

四、探究方法與驗證步驟

首先查詢網路資料，2014，提及未刪除「無效」的情況時，有 985824 種（如下圖），這是利用許志農（2024）計算排列的方法：從 $1, 2, 3, \cdots, 9$ 選取 4 個以上的數字進行直線排列，其情況數為 $P_4^9+P_5^9+P_6^9+P_7^9+P_8^9+P_9^9=985824$ 。

下圖為未刪除「無效」密碼的圖形密碼組合數。資料來源：擷取（David，2014）

```
In[1]:= allPermutations = Permutations[Range[1, 9], {4, 9}]

Out[1]= A very large output was generated. Here is a sample of it:

{{1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 3, 6}, {1, 2, 3, 7},
 {1, 2, 3, 8}, {1, 2, 3, 9}, {1, 2, 4, 3}, {1, 2, 4, 5}, {1, 2, 4, 6},
 {1, 2, 4, 7}, {1, 2, 4, 8}, <<985803>>, {9, 8, 7, 6, 5, 3, 2, 1, 4},
 {9, 8, 7, 6, 5, 3, 2, 4, 1}, {9, 8, 7, 6, 5, 3, 4, 1, 2}, {9, 8, 7, 6, 5, 3, 4, 2, 1},
 {9, 8, 7, 6, 5, 4, 1, 2, 3}, {9, 8, 7, 6, 5, 4, 1, 3, 2}, {9, 8, 7, 6, 5, 4, 2, 1, 3},
 {9, 8, 7, 6, 5, 4, 2, 3, 1}, {9, 8, 7, 6, 5, 4, 3, 1, 2}, {9, 8, 7, 6, 5, 4, 3, 2, 1}}

Show Less  Show More  Show Full Output  Set Size Limit...
```

圖形密碼鎖的規則

1、連續限制：一筆畫完，不可中斷。

2、數量限制：至少四個點，最多九個點。

3、跨越性：不一定要連到相鄰的點，只要兩點之間沒有其他點就可以相連。

4、交叉性：連線可以交叉。

5、消失性：若某個點已經連過，則該點會消失。

【$1\times n$ 情況探討】

1	2	3	4	5

以下為考慮上圖的5個數字全用的情況數。

考慮1□□□□：$[2],[3],[4],[5]$順序不變，有C_0^4種排列法，如下。

$5,(4),(3),(2),(1)$

考慮2□□□□：$[3],[4],[5]$順序不變，(1)任意放，有C_1^4種排列法，如下。

$2,(1),[3],[4],[5]$	$2,[3],(1),[4],[5]$	$2,[3],[4],(1),[5]$	$2,[3],[4],[5],(1)$

考慮3□□□□：$[4],[5]$順序不變，$(2),(1)$順序不變，有C_2^4種排列法，如下。

$3,(2),(1),[4],[5]$	$3,(2),[4],(1),[5]$	$3,(2),[4],[5],(1)$
$3,[4],(2),(1),[5]$	$3,[4],(2),[5],(1)$	$3,[4],[5],(2),(1)$

考慮4□□□□：$[5]$任意放，$(3),(2),(1)$順序不變，有C_3^4種排列法，如下。

$4,(3),(2),(1),[5]$	$4,(3),(2),[5],(1)$	$4,(3),[5],(2),(1)$	$4,[5],(3),(2),(1)$

考慮5□□□□：$(4),(3),(2),(1)$順序不變，有C_4^4種排列法，如下。

$5,(4),(3),(2),(1)$

因此以1×5為例，

恰使用 2 個數字，有$4\times\left(C_0^1+C_1^1\right)=8$種排列法。

恰使用 3 個數字，有$3\times\left(C_0^2+C_1^2+C_2^2\right)=12$種排列法。

恰使用 4 個數字，有$2\times\left(C_0^3+C_1^3+C_2^3+C_3^3\right)=16$種排列法。

恰使用 5 個數字，有$1\times\left(C_0^4+C_1^4+C_2^4+C_3^4+C_4^4\right)=16$種排列法。

由陳界山（2019）二項式定理可推得$C_0^n+C_1^n+C_2^n+\cdots+C_n^n=2^n$，

因此$1\times n$的密碼，可能情況總共有$2^1(n-1)+2^2(n-2)+2^3(n-3)+\cdots+2^{n-1}\times1=2^n-2n-2$種。

【2×2情況探討】不會三數共線，因此總共有$P_2^4+P_3^4+P_4^4=12+24+24=60$種可能性。

【2×3情況探討】如下，總共有$26+92+252+504+504=1378$種可能性。

1	2	3
4	5	6

恰使用 2 個數字，有$P_2^6-4=26$種可能性。【扣除$1\leftrightarrow3$、$4\leftrightarrow6$】

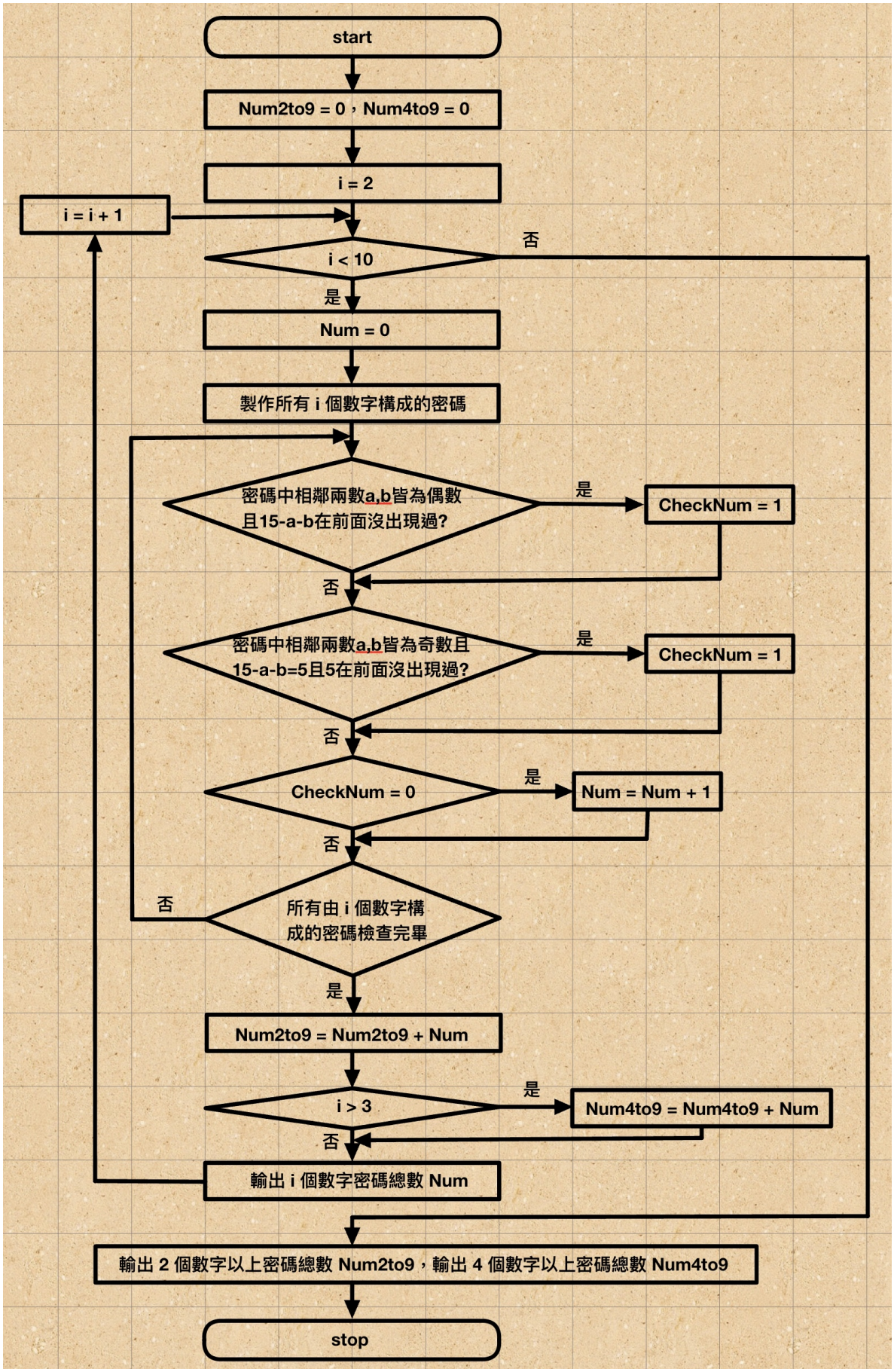
恰使用 3 個數字，有$P_3^6-(8+6)\times2=92$種可能性。【扣除錯誤連線方法】

恰使用 4 個數字，有$P_4^6-(24+18+12)\times2=252$種可能性。【扣除錯誤連線方法】

恰使用 5 個數字時，若將剩下的數字也連起來，就是恰使用 6 個數字的方法。

恰使用 6 個數字，有$6!-5!-5!+4!=504$種可能性。【扣除錯誤連線方法】

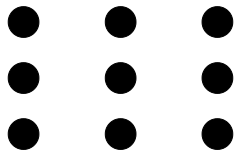
【3×3 情況探討】


start
Num2to9 = 0，Num4to9 = 0
i = 2
i = i + 1
i < 10
否
是
Num = 0
製作所有 i 個數字構成的密碼
密碼中相鄰兩數a,b皆為偶數
且15-a-b在前面沒出現過?
是
CheckNum = 1
否
密碼中相鄰兩數a,b皆為奇數且
15-a-b=5且5在前面沒出現過?
是
CheckNum = 1
否
CheckNum = 0
是
Num = Num + 1
否
否
所有由 i 個數字構
成的密碼檢查完畢
是
Num2to9 = Num2to9 + Num
i > 3
是
Num4to9 = Num4to9 + Num
否
輸出 i 個數字密碼總數 Num
輸出 2 個數字以上密碼總數 Num2to9，輸出 4 個數字以上密碼總數 Num4to9
stop

Android 手機的3×3圖形化螢幕鎖如圖一。

將 9 點轉換成數字的一般轉換法如圖二。

將 9 點轉換成數字的幻方轉換法如圖三。



圖一

1	2	3
4	5	6
7	8	9

圖二

2	9	4
7	5	3
6	1	8

圖三

幻方轉換法中，三數共線時其和必定為 15，且三數之和為 15 時必定共線。

因此由一一對應原理可知，檢驗三數是否共線，可轉換成檢驗三數之和是否為 15。

而三數共線的兩端可以分成下面兩種情況：

1、兩端皆為偶數。

2、兩端皆為奇數（此時中間的數字必定為 5）。

檢查規則 1：在兩端 A , B 皆為偶數時，從 A 連到 B 要先用掉15 - A - B 。

檢查規則 2：在兩端 A , B 皆為奇數且15 - A - B = 5 時，從 A 連到 B 要先用掉 5。

只要通過上面兩個檢查該畫法就是正確的，幻方轉換法大幅減少了檢查的步驟及數量，用程式進行檢查能在短時間內得到結果，此為本研究獨到之處。

【子程式 1】定義函式 ListPW(n,m) 的輸出結果為從1,2,3,⋯,n 取 m 個數字排列的所有情況。

```
import itertools
def ListPW(n,m):
  ListN=[]
  for i in range(1,n+1):
    ListN.append(i)
  ListM=list(itertools.permutations(ListN,m))
  return ListM
print(ListPW(5,2))
```

例如： ListPW(3,2) 為從1,2,3 取 2 個數字排列的所有情況，輸出結果如下：

[(1,2),(1,3),(2,1),(2,3),(3,1),(3,2)]

【子程式 2】定義函式 Check(Tuple,n) 為檢查畫法是否有錯，當輸出結果為 0 表示沒有錯誤，當輸出結果為1表示有錯誤。

```
def Check(Tuple,n):
  CheckNum=0
  for i in range(n-1):
    ListUsed=[]
    for j in range(i):
      ListUsed.append(Tuple[j])
    PW=15-Tuple[i]-Tuple[i+1]
    if Tuple[i]%2==0 and Tuple[i+1]%2==0 and PW not in ListUsed:
      CheckNum=1
    if Tuple[i]%2==1 and Tuple[i+1]%2==1 and PW==5 and 5 not in ListUsed:
      CheckNum=1
  return CheckNum
```

以下為了簡要說明，稱 Tuple[i]為 A， Tuple[i+1]為 B，15-A-B 為 PW。

第八行 if Tuple[i]%2==0 and Tuple[i+1]%2==0

這段在檢查：若 A , B 都是偶數，則 PW 要在 A 的前面就出現過（已經用掉）。

如果沒出現過，則 CheckNum =1（該畫法有錯誤）。

第十行 if Tuple[i]%2==1 and Tuple[i+1]%2==1 and PW==5

這段在檢查：若 A , B 都是奇數且 PW 是 5，則 5 要在 A 的前面就出現過（已經用掉）。

如果沒出現過，則 CheckNum =1（該畫法有錯誤）。

【主程式】利用函式 ListPW(n,m)生成所有畫法，再將畫法放入第二個函式 Check(Tuple,n)中檢查。每次通過檢查，則將正確畫法的數量+1。

```
Num2to9=0;Num4to9=0
for i in range(2,10):
  Num=0
  for TuplePW in ListPW(9,i):
    if Check(TuplePW,i)==0:
      Num=Num+1
  Num2to9=Num2to9+Num
  if i>3:
    Num4to9=Num4to9+Num
  print(i,'個點的密碼恰有',Num,'種')
print('四點至九點的密碼總共有',Num4to9,'種')
print('兩點至九點的密碼總共有',Num2to9,'種')
```

執行結果如下，歷時 10 秒。2 個點的密碼恰有 56 種，3 個點的密碼恰有 320 種，
4 個點的密碼恰有 1624 種，5 個點的密碼恰有 7152 種，6 個點的密碼恰有 26016 種，
7 個點的密碼恰有 72912 種，8 個點的密碼恰有 140704 種，9 個點的密碼恰有 140704 種。
四點至九點的密碼總共有 389112 種，兩點至九點的密碼總共有 389488 種。

五、結論與生活應用

完成 3×3 圖形鎖研究後，嘗試處理 3×4 與 4×4 圖形鎖，發現 3×4 圖形鎖除了要檢查左邊的 3×3 與右側的 3×3，還要檢查【左上↔右上】、【左中↔右中】、【左下↔右下】，通過以上檢查，才會是正確密碼。4×4 圖形鎖就更加複雜，以上這些是往後持續研究的主要課題。另外，這個手法也可以用來設計 OX 棋的 AI 對弈。

參考資料

1、書面資料

(1) 許志農（主編）（2024）。【SUPER】數學 2 講義。龍騰文化。

(2) 陳界山（主編）（2019）。數學（二）。南一書局。

2、網路資料

(1) 蘇聖雄（無日期）。手機的安全分析。2024 年 12 月 2 日。
https://my.stust.edu.tw/sys/read_attach.php?id=1029036

(2) David（2014 年 8 月 12 日）。Android lock password combinations。
https://reurl.cc/xp686N