

## 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：** 麻將小白的胡牌之路

### 一、摘要

麻將是一個具有歷史且富含變化的桌上型遊戲，不僅老少皆宜，更能夠增進人與人之間的互動與聯繫。然而，對於麻將新手來說，入門雖然很容易，但是要獲勝確實有不少學習的門檻。本研究的目的，是希望藉由電腦輔助程式，協助麻將新手在遊戲的過程中，理解麻將每張牌組的抽牌機率並計算牌組中的留牌機率，以期大幅提升玩家勝率。這次的程式是藉由 C++ 編寫而成，目前程式最核心的技術，是探討牌型組合並運用多個巢狀迴圈，計算出不同手牌被抽到的機率，提供麻將新手自我訓練與學習參考。

### 二、探究題目與動機

作者選定麻將作為研究的主題，是因為它是一個非常熱門的遊戲，它的變化與排列組合遠比其他桌上型遊戲更豐富，在遊戲進行的過程中，氣氛不僅熱絡，更能維繫家人朋友間的感情。除此之外，麻將還有許多好處，包括訓練大腦（如何出牌、是否吃與碰）、時間相對其他遊戲短暫，節奏明快、沒有年齡限制，老少皆宜等，讓大家都能夠樂在其中。然而，也由於麻將的變化豐富，特別是對於新手來說，獲勝並不是那麼容易。因此，我想要將麻將與電腦程式互相結合，不僅熟悉程式設計邏輯，將電腦程式與麻將相結合後，也能藉此協助麻將新手增加獲勝的機率。

### 三、探究目的與假設

#### （一）研究目的

1. 理解新手常見的難題，思考藉由程式輔助學習。在有限的時間內，完成可執行的程式。
2. 設計麻將的基礎程式，自動產出四位玩家的手牌及桌上的棄牌，撰寫程式識別順子與坎子的各種組合，讓麻將新手可以查詢特定牌張的抽牌機率。
3. 設計麻將的進階程式(吃與碰策略組合分析)，可以提供麻將新手每張手牌留牌參考機率。

#### （二）研究假設

1. 麻將輔助程式的撰寫與操作是可行的。
2. 自動產出四位玩家的手牌及桌上的棄牌是可行的。
3. 這個麻將輔助程式可正確判別順子與坎子的組合。
4. 麻將新手可以利用這個程式，查詢特定牌張的抽牌機率。
5. 此程式可以初步估計順子與坎子的組合可能性、提供麻將新手留牌機率的參考。

### 四、探究方法與驗證步驟

#### （一）麻將小白好奇的問題與程式解決方案

表一：麻將小白關注的問題與思考脈絡

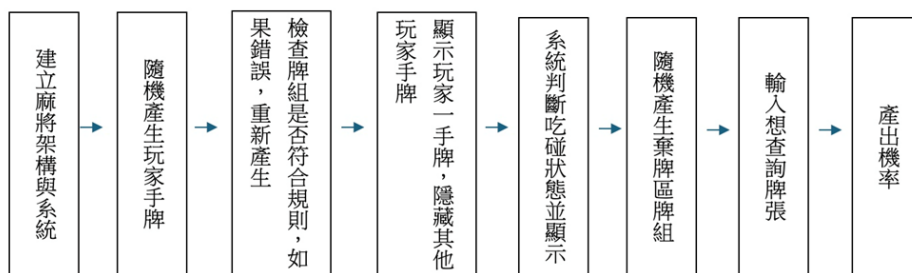
問題	思考脈絡
希望知道每張牌的抽牌機率	有機會解決(門檻較低)

希望知道每張牌的留牌的機率	有機會解決(門檻較高)
希望知道平均抽幾張就會有人胡牌	目前有兩個可行的做法。一：寫信給神來也麻將公司，希望取得遊戲資料庫，得知平均抽幾張牌就會有人胡牌。二：設計自動玩麻將的程式，讓玩家打一百萬局麻將，統計平均抽幾張就會有人胡牌。方法一未必可行，而且與程式無關。方法二雖然不錯，但短時間內無法完成。
希望知道特殊牌型的胡牌機率	同上。
傳說中麻將口訣都是真的嗎?	同上。
跟老手打牌一定會輸嗎?	欠缺麻將遊戲資料庫，設計自動麻將的程式不易。

### (二) 麻將小白的研究器材

1. 硬體: 筆記型電腦，配備: Intel Core 7
2. 軟體: DEV C++。C++是一個強大的系統，它能整理形形色色的文獻資料，並撰寫出實用的程式。比起市面上常見的 Python、Java 等系統更能幫助未來在資訊工程方面有顯著的成效，因為它是比較符合電腦的語言，所以更適合在執行專題的時候執行。

### (三) 研究思考脈絡流程圖



圖一：程式設計步驟圖

### (四) 研究進行步驟

1. 建立麻將程式的基礎程式 (包括卡牌張數、建立棄牌區等機制)
2. 自動產出玩家的手牌
3. 自動產出棄牌區的牌張
4. 自動校對產出的牌張是否符合麻將規則
5. 判讀順子與坎子的機制
6. 計算目前每張牌剩下的張數並根據剩下的可抽牌數，計算該張牌被抽到的機率。
7. 根據順子與坎子的組合機率，分析每張牌的留牌機率

### (五) 困難與解決問題的歷程

1. 如何建構麻將的牌組?

麻將由五種不同的花色組成，分別有萬、筒、索、字、花的種類。由於萬、筒、索皆由 1 至 9 號牌組合而成，而每一張牌又各有四張，所以我用  $4 * 9$  的二維陣列儲存它們，稱它們為

number card。字和花牌沒有順序，稱他們為 word card。其中花牌比較特別，只有春夏秋冬梅蘭竹菊八張，為了要和 7 種字牌湊成與萬、筒、索相同的 9 種面額，所以春夏秋冬會統一稱為「季」；而梅蘭竹菊則會統一稱為「花」。

```
string num_card[9]= {"一","二","三","四","五","六","七","八","九"};
string word_card[9]={ "東","西","南","北","中","發","白","花","季"};
string class_card[4]={ "筒","索","萬","字"};
//產生手牌
int new_Self[4][16]={}; //宣告新的手牌陣列，最多 4位玩家 16張手牌
int check_Card[4][9]={}; // 檢查各種花色 1-9 號牌 出現次數
int reveal_Card[4][9]={}; // 檯面 各種花色、號牌 出現張數
std::string strand;
```

## 2. 設置四位玩家的手牌以及自動產生玩家的手牌

先編寫一個 4\*16 的二維陣列，再藉由程式的 rand%變數，生成四位玩家的手牌。

```
int new_Self[4][16]={};

void auto_deal_card(){
    for(i=0;i<4;i=i+1) { //重置 各種花色手牌出現次數為0
        for(j=0;j<16;j=j+1){
            check_Card[i][j]=0; }}

    for(i=0;i<player;i=i+1) { //迴圈重複4(玩家人數)次
        for(j=0;j<card;j=j+1){ //迴圈重複16(手牌張數)次
            //給使用者的說明
            // cout << "產生手牌，第"<<i+1 << "位，第" << j+1<<"張=">;
            c1=rand()%4; c2=rand()%9; // 產生 C1:花色、C2:數字
```

## 3. 設置棄牌區以及自動產出棄牌區的牌張

藉由手動輸入棄牌區的總張數，讓電腦自動產生棄牌區的牌張。

```
void seat_card() {
    int i,j;
    cout << "\n\n請輸入棄牌區的張數 => "; //輸出棄牌區的張數
    cin >> seat_number; //讓使用者輸入棄牌區的張數
    for(i=0;i<seat_number;i=i+1){ //產生棄牌
        c1=rand()%4; c2=rand()%9; // C1:花色、C2:數字
        while(check_Card[c1][c2]>=4){ //檢查同花色手牌超過四張
            c1=rand()%4; c2=rand()%9; }
        check_Card[c1][c2]++;
```

## 4. 如何偵測不合理的牌張並重新組成

如果發現同種面額的牌張超過四張，系統就會自動照規則生成新的麻將手牌。

```
void auto_deal_card(){
    for(i=0;i<4;i=i+1) { //重置 各種花色手牌出現次數為0
        for(j=0;j<16;j=j+1){
            check_Card[i][j]=0; }}

    for(i=0;i<player;i=i+1) { //迴圈重複4(玩家人數)次
        for(j=0;j<card;j=j+1){ //迴圈重複16(手牌張數)次
            //給使用者的說明
            // cout << "產生手牌，第"<<i+1 << "位，第" << j+1<<"張=">;
            c1=rand()%4; c2=rand()%9; // 產生 C1:花色、C2:數字

            while(check_Card[c1][c2]>=4){ //檢查同花色牌出現超過四次
                c1=rand()%4; c2=rand()%9; //重複產生 C1:花色、C2:數字
                check_Card[c1][c2]++;
```

## 5. 如何讓玩家 1 顯示手牌，而剩下的玩家隱藏手牌

程式會讓玩家 1 的手牌全部顯示中文字，而讓其餘玩家的手牌大部分都顯示數字（除非他們有湊成吃或碰的牌型，才能顯示中文字）。

```
    }else{
        //顯示玩家1的手牌
        if(new_Self[i][j]>100){
            new_Self[i][j]-100;
            //顯示中文字
            c1=(new_Self[i][j]-100)/10; // 花色 0:筒·1:索·2:萬·3:字
            c2=(new_Self[i][j]-100)%10; // 數字 1-9
            if(c1==0 || c1==1 || c1==2){
                cout << " " << num_card[c2]; // 顯示 一、二、...九
            } else if(c1==3) {
                cout << " " << word_card[c2]; // 顯示 東、西、南、北
            }
        }else{
            c1=(new_Self[i][j])/10; // 花色 0:筒·1:索·2:萬·3:字
            c2=(new_Self[i][j])%10; // 數字 1-9
            if(c1==0 || c1==1 || c1==2){
                cout << " " << num_card[c2]; // 顯示 一、二、...九
            } else if(c1==3) {
                cout << " " << word_card[c2]; // 顯示 東、西、南、北
            }
        }
    }
}
```

## 6. 如何利用建構的程式，查詢某一張牌的抽牌機率→目的 1 解決

先將全部的麻將牌減四位玩家以及棄牌區的牌，再計算出分母與分子，最後程式使用除法計算該張牌被抽到的機率。

```
//計算機率的分子
whole = 144 - player*card - seat_number; //全部可以抽的牌
cout << "\n目前可能抽的牌等於" << whole << "張" << endl; //顯示幕前可抽牌的張數

//計算分子(棄牌區)
for(i=0;i<seat_number;i=i+1){ //宣告for迴圈的要件
    if(SEAT[i] == S) { //如果棄牌區有此面額的話
        Ccard = Ccard + 1; // · Ccard迴圈的數目會+1
    }
}
cout << "\n目前此牌已經在棄牌區出來 " << Ccard << "張了" << endl;

//計算分子(手牌區)
for(i=0;i<player;i=i+1){
    for(j=0;j<card;j=j+1){
        if(new_Self[i][j] == S+100){ //如果手牌區【非蓋牌】有此面額的話
            Ccard = Ccard + 1; // · Ccard迴圈的數目會+1
        }
        //玩家1全部統計
        if(new_Self[i][j] == S && i==0){ //玩家一【蓋牌】有此面額的話
            Ccard = Ccard + 1; // · Ccard迴圈的數目會+1
        }
    }
}
cout << "目前此面額已經出來 " << Ccard << "張了" << endl;

double son = 4-Ccard; //還有幾張目標牌可以抽
//計算抽牌的機率
double pro = son/whole;
std::cout << "目前抽到此牌的機率為" << pro << std::endl; //顯示剩餘此牌面額的張數

return 0;
```

## 7. 如何讓系統判斷已經是順子或坎子的組合，如果是玩家 2、3、4 有此牌型就開牌

先使用巢狀迴圈判斷玩家 2、3、4 的該張手牌已經形成順子或坎子的組合，再使用判別是判斷如果是此牌型，就將它顯示為中文字（其餘顯示數字）。

```

void show_card(){
    cout << endl;
    for(i=0;i<player;i=i+1) { // 顯示 4位玩家 手牌
        cout << "玩家[" << i+1 << "] 手牌-----" << endl;
        for(j=0;j<card;j=j+1){ // 每位玩家迴圈重複16(手牌張數)次
            if(i != 0){ // 別的玩家的牌都必須部份顯示文字
                if(new_Self[i][j]<100) {
                    cout << " " << new_Self[i][j]+1 ;//顯示數字
                } else {
                    //顯示中文字
                    c1=(new_Self[i][j]-100)/10; // 花色 0:筒、1:索、2:萬、3:字
                    c2=(new_Self[i][j]-100)%10; // 數字 1-9
                    if(c1==0 || c1==1 || c1==2){
                        cout << " " << num_card[c2]; // 顯示 一、二、...九
                    } else if(c1==3) {
                        cout << " " << word_card[c2];// 顯示 東、西、南、北
                    }
                }
            }
        }
    }
}

```

### 8. 如何讓系統判斷玩家 1 手上有機會組成順子或坎子的牌型

使用 if 判別式判斷兩張之間如果是絕對值 1 或 2 的面額，就宣告將會湊成順子；而兩張面額相同的手牌，則會被宣告將會組成坎子。

```

if(reserve[i]<100 && reserve[i]<30) { // <30 代表 筒、索、萬牌、可以 碰牌+吃牌
    c1=reserve[i]/10;
    c2=reserve[i]%10;
    if(reserve[i-1]==reserve[i]) { // 左碰
        prob_reserve[i]=prob_reserve[i]+(4-reveal_Card[c1][c2]);
        if(reserve[i-2]-reserve[i]==1) { // 左 4索 3索 *3索*
            if(c2==8) { // 1萬 9索 *9索*(20 18 18) 無法左碰吃
                prob_reserve[i]=prob_reserve[i];
            }
            } else if(c2==7) { // 9索 8索 *8索* -->右吃機率
                prob_reserve[i]=prob_reserve[i]+(4-reveal_Card[c1][c2-1]);
            } else if(c2==0) { // 2索 1索 *1索* -->左吃機率
                prob_reserve[i]=prob_reserve[i]+(4-reveal_Card[c1][c2+2]);
            } else { // 4索 3索 *3索* -->左吃機率+右吃機率
                prob_reserve[i]=prob_reserve[i]+(4-reveal_Card[c1][c2+2])+(4-reveal_Card[c1][c2-1]);
            }
        }
        } else if(reserve[i-2]-reserve[i]==2) { //左碰吃 5索 3索 *3索*
            if(c2!=8 && c2!=7) { // 無法吃 2萬 9索 *9索*(21 18 18)* 1萬 8索 *8索*(20 17 17)
                prob_reserve[i]=prob_reserve[i]+(4-reveal_Card[c1][c2+1]);
            }
        }
    }
}

```

### 9. 如何讓系統依照上述條件，計算出手上每張牌的留牌機率

使用 for 迴圈的功能，讓程式計算並輸出手上每張牌的留牌機率。

```

cout << "\n\n玩家【1】留牌價值統計----";
for(i=2; i<=17; i++) { cout<<prob_reserve[i] << " "; }

```

### (六) 研究結果：

1. 可以自動產出四位玩家的手牌。
2. 可以自動顯示玩家 1 的手牌（以文字呈現），將其他三位玩家蓋牌（以數字呈現，已經成為順子、坎子的以文字呈現）。

請輸入玩家的名稱=>麻將小白

玩家【1】手牌	北	西	西	五	四	二	二	八	八	七	五	一	五
季 季 北	37	37	36	23	23	18	13	11	9	5	5	4	4
季 季 北	37	37	36	23	23	18	13	11	9	5	5	4	4
季 季 北	39	37	32	31	31	27	26	23	19	15	14	14	5
季 季 北	39	37	32	31	31	27	26	23	19	15	14	14	5
季 季 北	37	34	31	27	24	24	23	21	17	14	11	7	6
季 季 北	37	34	31	27	24	24	23	21	17	14	11	7	6

3. 可以自動輸入棄牌區的張數，並讓程式隨機產生棄牌區的牌張。

```
請輸入棄牌區的張數 => 20
一筒 二筒 九筒 二萬 九筒 七筒 花 西 一萬 北 七萬 一筒 一索 七筒 發 九萬 六筒 三索 七萬 七筒
```

4. 系統會根據玩家的手牌以及棄牌區的張數，統計每張牌出現的次數。

```
檯面及手牌出現的張數統計----
筒 3 2 1 0 1 1 3 0 2
索 2 0 1 1 2 1 2 3 1
萬 1 3 0 1 1 0 2 0 1
字 0 3 0 3 0 1 0 1 2
```

5. 輸入想要查詢的牌張，並依照現有的剩下牌張，計算該牌張被抽到的機率。

```
請輸入您要搜尋的卡(0:筒、1:索、2:萬)=>1
請輸入您要搜尋的卡(數字請用阿拉伯數字)=>8

您要搜尋的是： 八索
目前可能抽的牌等於60張

目前此牌已經在棄牌區出來 0張了
目前此面額已經出來 3張了
目前抽到此牌的機率為0.0166667
```

6. 系統會根據玩家 1 的手牌，分析將來吃或碰的可能性，決定每張牌的留牌價值。分數越高者，留牌價值越高。

```
玩家【1】手牌-----
季 季 北 北 西 西 五 四 二 二 八 八 七 五 一 五
萬 萬 萬 萬 索 索 索 索 索 筒
```

```
玩家【1】留牌價值統計----2 2 1 1 1 1 8 12 5 5 7 7 9 3 0 0
```

## 五、結論與生活應用

這個程式能幫助麻將小白更輕鬆的判斷抽牌或留牌的機率，輔助他們在遊戲進行中更知道如何打牌，以提高獲勝的機會。除此之外，也使用隨機生成的迴圈，產生四位玩家的手牌；並藉由文字與數字的方式讓操作者辨別這些手牌是否湊成吃與碰的組合（已經湊成吃或碰的手牌以文字呈現，玩家 1 除外）。本程式也分析了多種湊成順子的牌型，讓新手玩家能夠大開眼界，並知道麻將的吃碰不是只有單純的「三張牌組合」。

本研究未來目標，希望能將該程式與視覺分析軟體結合，不管是實體還是線上麻將，都能自動擷取其中訊息並進行分析。而長期目標，希望利用資料庫的資訊，與 AI 功能結合，以分析麻將進行過程的動態資訊，提供麻將小白出牌和留牌的即時訊息，最大化獲勝的機率。

## 參考資料

1. 麻將尊者(2006)。台灣麻將高手。知青頻道。
2. 洪錦魁(2017)。邁向賭神之路：麻將必勝秘笈。上奇時代。