

Java 期末-猜數字遊戲

S0961118 資管一 陳羿華

一、想法

先取得需要的東西

```
public class ABullsCows extends BasicBullsCows
{
    int[][] set;           可能答案集
    int[][] set1;

    public int [] GenerateInputNumber()
    {
        //access game record***** Total Round *****
        int RoundNumber = this.GetRoundNumber();
        int [][] Inputs = this.GetInputs();
        int [] Bulls = this.GetBulls(); 取得答案、幾 A、幾 B
        int [] Cows = this.GetCows();
        int [] candidateNumber = new int[4]; 將回傳的答案

        //print
        for(int i = 0; i < RoundNumber; i++)
        {
            System.out.print("Round " + i + ": "
                + Inputs[i][0] + Inputs[i][1] + Inputs[i][2] + Inputs[i][3]
                + " " + Bulls[i] + "A" + Cows[i] + "B\n");
        }
    }
}
```

1. 找出全部可能的答案，並放入一個二維陣列中(set)，共 $9*9*8*7=4536$ 個

```
//all possible answers
if(RoundNumber==0)
{
    set=new int[4536][4]; 全部可能答案有 4536 個，每個答案有 4 個數字
    int i=0;

    for(int n1=0;n1<=9;n1++)
    {
        if(n1==0) continue;
        for(int n2=0;n2<=9;n2++)
        {
            if(n1==n2) continue;
            for(int n3=0;n3<=9;n3++)
            {
                if(n3==n1 || n3==n2) continue;
                for(int n4=0;n4<=9;n4++)
                {
                    if(n4==n1 || n4==n2 || n4==n3) continue;
                    set[i][0]=n1;
                    set[i][1]=n2;
                    set[i][2]=n3;
                    set[i][3]=n4;
                    i++;
                }
            }
        }
    }
}
```

2. 從陣列(set)中隨機挑一個當第一次猜測

```
//the first guess
candidateNumber=set[(int)(Math.random()*4536)];
}
}
```

3. 猜完後會出來幾 A 幾 B。將第一次猜測與陣列中其他數字比對，找出同樣也是幾 A 幾 B 的

```
if(RoundNumber>0)
{
    int a=Bulls[RoundNumber-1];
    int b=Cows[RoundNumber-1];
    int Round1=0; 對比幾 A 幾 B 後，剩餘的答案個數

    set1=new int[4536][4]; 建新陣列，放入剩餘可能答案

    for(int j = 0; j < 4536; j++)
    {
        int Newbulls = 0;
        int Newcows= 0;
        //match 1st
        for(int k=0;k<=3;k++)
        {
            if(Inputs[RoundNumber-1][0] == set[j][k] )
            {
                if(k == 0)
                    Newbulls++;
                else
                    Newcows++;
                break;
            }
        }
    }
}
```

```
//match 2nd
for(int k = 0; k <= 3; k++)
{
    if( Inputs[RoundNumber-1][1] == set[j][k] )
    {
        if(k == 1)
            Newbulls++;
        else
            Newcows++;
        break;
    }
}

//match 3th
for(int k = 0; k <= 3; k++)
{
    if( Inputs[RoundNumber-1][2] == set[j][k] )
    {
        if(k == 2)
            Newbulls++;
        else
            Newcows++;
        break;
    }
}

//match 4th
for(int k = 0; k <= 3; k++)
{
    if( Inputs[RoundNumber-1][3] == set[j][k] )
    {
        if(k == 3)
            Newbulls++;
        else
            Newcows++;
        break;
    }
}
```

4. 將同樣也是幾 A 幾 B 的結果放入另一個二維陣列(set1)

```
if(Newbulls==a&&Newcows==b)
{
    set1[Round1][0]=set[j][0];
    set1[Round1][1]=set[j][1];
    set1[Round1][2]=set[j][2];
    set1[Round1][3]=set[j][3];
    Round1++;
}
```

5. 將 set 初始化，並把第二個陣列(set1)寫入 set 中

```
set=new int[Round1][4];
set=set1;
```

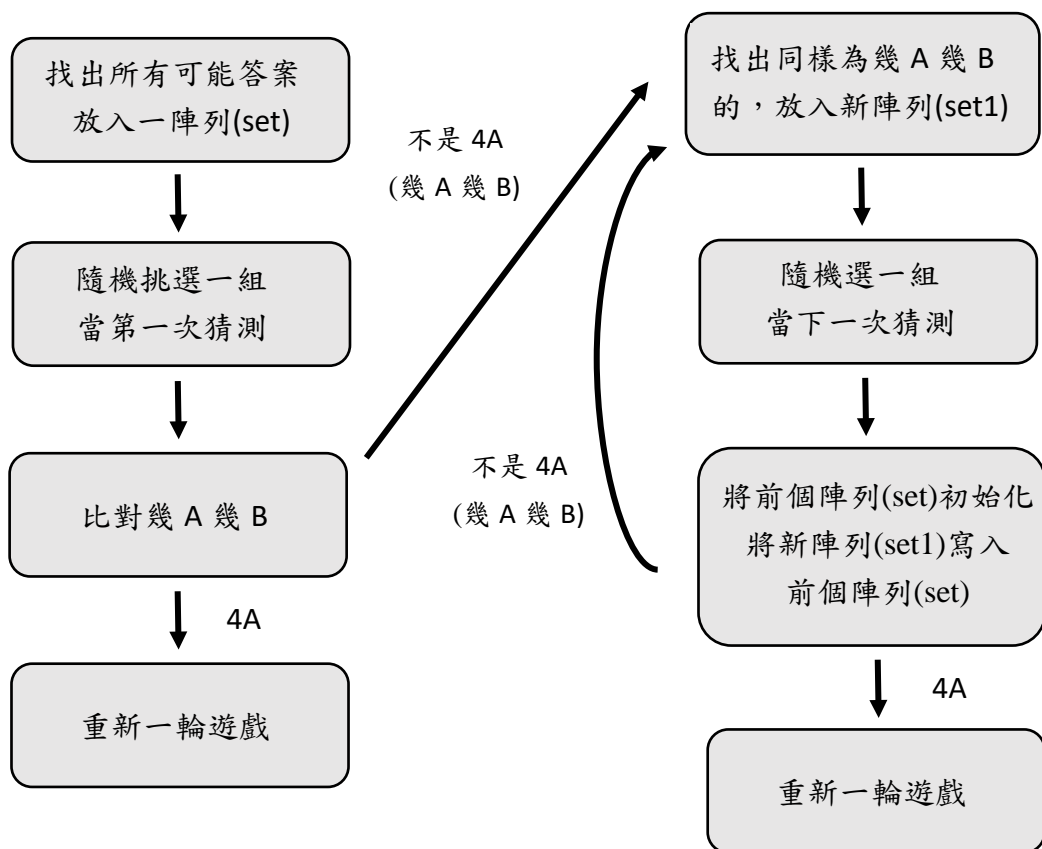
6. 從陣列中(set1)隨機挑一個當下一次猜測，若挑到不符合 AB 遊戲規則的數字 (因 set1 長度大於找出的數字個數，因此可能挑到 0000)，則重挑

```
set=new int[Round1][4];
set=set1;
boolean passCheck = false;
while(passCheck == false)
{
    candidateNumber=set1[(int)(Math.random()*(Round1))];
    passCheck = CheckInputNumber(candidateNumber);
}

return candidateNumber;
```

回傳答案

7. 重複 3.4.5.6 直到猜出答案



二、 平均的猜題回合數

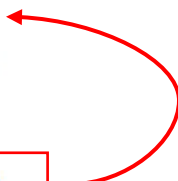
平均 : 5 點多次 (跑多次確認每次結果皆為如此)

```
The average rounds of your Bulls and Cows game = 5.438  
BUILD SUCCESSFUL (total time: 1 second)  
  
The average rounds of your Bulls and Cows game = 5.377  
BUILD SUCCESSFUL (total time: 1 second)  
  
The average rounds of your Bulls and Cows game = 5.409  
BUILD SUCCESSFUL (total time: 1 second)  
  
The average rounds of your Bulls and Cows game = 5.378  
BUILD SUCCESSFUL (total time: 1 second)  
  
The average rounds of your Bulls and Cows game = 5.445  
BUILD SUCCESSFUL (total time: 1 second)
```

三、 編寫時遭遇的錯誤

1. 原本將陣列直接初始化寫在程式一開始，導致猜下一個答案時，前一次寫入的資料全不見

```
//all possible answers  
if(RoundNumber==0)  
{  
    set=new int[4536][4];  
    int i=0;
```



2. 忽略新建的陣列(set1)比可能答案的個數多，跑下一次時，沒放入值的部分(0000)比對完為 0A0B 會被放入新陣列。隨機選時會被選到，因此加了確認答案是否符合遊戲規則這步驟，若不符合則重挑

```
boolean passCheck = false;  
while(passCheck == false)  
{  
    candidateNumber=set1[(int)(Math.random()*(Round1))];  
    passCheck = CheckInputNumber(candidateNumber);  
}
```


四、 遊戲過程片段

```
Round 0: 2167 1A0B
remaining possible answers:420
```

```
-----
Round 0: 2167 1A0B
Round 1: 5148 1A1B
remaining possible answers:78
```

```
-----
Round 0: 2167 1A0B
Round 1: 5148 1A1B
Round 2: 4130 1A1B
remaining possible answers:15
```

```
-----
Round 0: 2167 1A0B
Round 1: 5148 1A1B
Round 2: 4130 1A1B
Round 3: 9105 0A2B
remaining possible answers:2
```

```
-----
Round 0: 2167 1A0B
Round 1: 5148 1A1B
Round 2: 4130 1A1B
Round 3: 9105 0A2B
Round 4: 2540 1A2B
remaining possible answers:1
```

```
Round 0: 7296 1A1B
remaining possible answers:660
```

```
-----
Round 0: 7296 1A1B
Round 1: 7039 0A0B
remaining possible answers:48
```

```
-----
Round 0: 7296 1A1B
Round 1: 7039 0A0B
Round 2: 6284 3A0B
remaining possible answers:4
```

```
-----
Round 0: 7296 1A1B
Round 1: 7039 0A0B
Round 2: 6284 3A0B
Round 3: 6214 3A0B
remaining possible answers:1
```