

# 商管程式設計 (109-1)

## 作業八

作業設計：孔令傑  
國立臺灣大學資訊管理學系

繳交作業時，請至 PDOGS (<http://pdogs.ntu.im/judge/>) 為第一、二題各上傳一份 Python 3.5 原始碼 (以複製貼上原始碼的方式上傳)。每位學生都要上傳自己寫的解答。不接受紙本繳交；不接受遲交。

這份作業的截止時間是 **2020 年 12 月 9 日晚上九點**。在你開始前，請閱讀課本的第十四章一至四節，以及第十五至十七章<sup>1</sup>。為這份作業設計測試資料並且提供解答的助教是鄭宇伶。

**小叮嚀：**請注意 PDOGS 上的 python 版本是 3.5，此版本的 dictionary 不像後面的版本會自動排序，若是同一程式碼每一次上傳結果都不一樣，有可能是這個問題。

### 第一題

(10 分) 大家小時候應該都有看過撲克牌，或甚至玩過一些撲克牌遊戲，在某些撲克牌遊戲中，例如梭哈、大老二、十三張、德州撲克等等，都會以五張牌的組合，比較大小來決定勝負。而每張撲克牌都有特定的花色和點數，這樣的結構讓我們很自然地會考慮把撲克牌包成類別。由於前幾週的課程介紹過類別，在本題就讓我們來練習設計和實作類別吧<sup>2</sup>！我們將在程式中建立名為 `Card` 的類別，每個 `Card` 物件裡面儲存該張撲克牌的花色和點數。類別的一部份如下：

```
class Card:
    def __init__(self, suit, rank):
        self.suit = suit
        self.rank = rank

    # something else
```

本題會發給你五張撲克牌，要求你決定如何出牌。你的程式應該讀取撲克牌資訊，建立五個 `Card` 物件，並依照以下規則計算分數：

- 規則 a：如果你打出一張 A，你可以獲得 1 分。
- 規則 b：如果你打出一對，也就是有兩張相同點數的牌，每對可以獲得 2 分。
- 規則 c：如果你打出同花，也就是五張同一花色的牌，你可以獲得 3 分。
- 規則 d：如果你打出順子，也就是五張順連的牌，你可以獲得 5 分。順連與否不考慮花色，也不因點數而中斷，比如說黑桃 Q、梅花 K、方塊 A、梅花 2、黑桃 3 這種組合也算順連。換言之，不考慮花色的話，共有 13 種順連的方式。
- 規則 e：如果你打出葫蘆，也就是三張同一點數的牌，加一對其他點數的牌，你可以獲得 10 分。
- 規則 f：如果你打出四條，也就是四張同一點數的牌，你可以獲得 20 分。

<sup>1</sup>課本是 A. Downey 所著的 *Think Python 2*，在 <http://greenteapress.com/wp/think-python-2e/> 可以下載。

<sup>2</sup>以本題的情境和任務來說，當然沒有非得要設計和實作類別不可，但就當作是教學目的，請大家還是乖乖跟著練習吧。

- 規則 g：如果你打出同花順，也就是五張同一花色且順連的牌，你可以獲得 100 分。

在遊戲中，一張牌只能用一次。比如說當你能打出同花順，代表你一定也可以打出同花或是順子，這時候你只會獲得同花順的分數（或者說，我們假設你會選擇以分數最大化的方式出牌）；又或者是你打出了四張 7、一張黑桃 A，這時你會獲得四條以及一張 A 的分數，共 21 分，一對不重複計分。

你可能會想在類別中建立某些 method（成員函數），比如判斷該張牌的點數是否為 A，並回傳布林值。此外，你從上述規則應該可以知道，同樣的五張牌可能有多種不同的組合，獲得的分數也可能不同，而你或許可以寫一個 method 去計算能獲得的最高分數。請找出可以獲得最高分數的出牌方式，並印出該分數。

## 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有 2 列。第一列為五個字元  $S_1$ 、 $S_2$  直到  $S_5$ ，依序代表五張牌的花色，其中  $S_i \in \{S, H, D, C\}$ ， $\{S, H, D, C\}$  依序代表是黑桃、紅心、方塊以及梅花。第二列為五個字元  $R_1$ 、 $R_2$  直到  $R_5$ ，依序代表五張牌的點數，其中  $R_i \in \{A, 2, 3, \dots, J, Q, K\}$ 。已知不會有兩張一模一樣的撲克牌出現，且每一行的任兩個字元之間被一個逗號隔開。請印出這副牌所能獲得最高的分數。

舉例來說，如果輸入是

```
H,D,S,C,S
A,A,A,8,8
```

則輸出應該是

```
10
```

如果輸入是

```
C,S,D,H,D
A,2,A,8,8
```

則輸出應該是

```
4
```

如果輸入是

```
C,S,D,H,D
A,A,A,8,8
```

則輸出應該是

```
10
```

如果輸入是

```
D,H,S,C,C
J,5,J,J,2
```

則輸出應該是

2

## 你上傳的原始碼裡應該包含什麼

你的.py 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 Python 3.5 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。本題共有 5 組測試資料，一筆測試資料佔 2 分。

## 第二題

(20 分) 接續第一題，現在有一群人要一起玩這個撲克牌遊戲，每個人將會拿到五張牌。一個人所拿到的牌組叫做 **Deck**，一個 **Deck** 中會有許多張 **Card**。你或許會想要再建立名為 **Deck** 的類別，每個 **Deck** 物件裡面儲存名為 **cards** 的卡片清單，清單中的每個元素都是一個 **Card** 物件，而 **Deck** 類別的一部份如下：

```
class Deck:
    def __init__(self):
        self.cards = []

    def add_card(self, card):
        self.cards.append(card)

    # something else
```

你的主程式應該建立一些 **Deck** 物件，並在讀到新卡片的時候建立一個 **Card** 物件，再放到某個 **Deck** 物件中。在本題中，一個人拿到的牌並不會重複。但為了增加遊戲的困難度，我們新增了一個規定，每個人在開局時會依序收到九張牌，若新拿到一張牌，其點數為手中某張牌的點數加一，且兩張牌花色相同，那麼這兩張牌就必須被拿掉，須注意點數 10、J、Q、K、A 的點數加一依序為點數 J、Q、K、A、2。一直拿到九張牌都看過了，或是拿滿五張牌了。舉例來說，若你拿到了黑桃 2、方塊 3、黑桃 5，新拿到一張方塊 4，此時方塊 3 和方塊 4 都必須被拿掉。再新拿到一張紅心 8，可以繼續拿牌，再新拿到一張黑桃 6，此時黑桃 5 和黑桃 6 都必須被拿掉。

你可能會想幫你的類別實作一些成員函數，例如在 **Deck** 類別中寫一個 **method**，確保新增加的卡片沒有違反新規定，又或者是回傳牌組總分。若你覺得有幫助，你可以寫更多 **method**。你也可以參考第一題所寫的成員函式，並在本題加以擴充。

本題會給定人數和若干組撲克牌，請計算並印出每一個人能拿到的最高分數。

## 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，會有  $n + 1$  列。第一列為一個數字  $n$ ，代表本次有幾個人參與撲克牌遊戲，其中  $n \in \{1, \dots, 10\}$ ；第二到  $n + 1$  列分別有九張牌的資訊，第  $i + 1$  列依序是  $C_{i,1}$ 、 $C_{i,2}$  到  $C_{i,9}$ ，兩兩皆以一個逗點隔開，其中  $C_{ij}$  有兩個字元，分別為  $S_{i,j}$  及  $R_{i,j}$ ， $S_i \in \{S, H, D, C\}$ 、 $R_i \in \{A, 2, 3, \dots, J, Q, K\}$ ， $\{S, H, D, C\}$  依序代表是黑桃、紅心、方塊以及梅花。已知同一列中不會有兩張一模一樣的撲克牌出現。請依序印出每個人所能獲得最高的分數，兩兩以一個逗號隔開。

舉例來說，如果輸入是

```
2
SK,S8,S3,H2,C6,SA,S2,H3,S4
SK,S8,S3,H2,S4,SA,S2,H3,C6
```

則輸出應該是

```
0,0
```

請注意按照遊戲規則，一號玩家一拿到梅花 6 的時候他就必須出牌了，因此後面的黑桃 A、黑桃 2、愛心 3 和黑桃 4 都不屬於他，而二號玩家拿到黑桃 4 時要把黑桃 4 和已經在手的黑桃 3 一起丟掉，接著拿到黑桃 A 時，要把黑桃 A 和已經在手的黑桃 K 一起丟掉，然後拿到黑桃 2，再接著拿到愛心 3 時，要把愛心 3 和已經在手裡的愛心 2 一起丟掉，最後拿到梅花 6，所以二號玩家手中所有的牌分別為黑桃 8、黑桃 2 以及梅花 6，此時沒有可以出的牌，二號玩家得分為 0。如果輸入是

```
3
SA,H9,H3,D4,SJ,CA,HQ,D8,C7
SJ,S10,HK,S5,D5,SK,H6,DQ,C8
CA,D7,H9,C6,S9,D6,H10,D10,HA
```

則輸出應該是

```
1,2,3
```

如果輸入是

```
4
SJ,DJ,C6,C10,CQ,S5,CA,S2,D9
S6,D8,H2,SJ,S2,D6,D7,HQ,C5
C8,SK,H6,H7,DJ,SJ,D8,D6,HJ
HA,C2,SK,SQ,CJ,D5,H2,S4,C9
```

則輸出應該是

```
2,2,4,5
```

## 你上傳的原始碼裡應該包含什麼

你的.py 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 Python 3.5 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 評分原則

這一題的所有分數都根據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。本題共有 10 組測試資料，一筆測試資料佔 2 分。

## 第三題

(30 分) 艾莎很喜歡狗狗，她不定期地會去動物之家領養狗狗，平常會帶狗狗一起散步，偶爾幫狗狗洗澡，為了狗狗的健康著想，艾莎會做狗狗的活動紀錄，希望可以透過紀錄追蹤狗狗的狀況。

艾莎會依據累積的灰塵量決定是否要幫狗狗洗澡，狗狗在被領養的當天灰塵累積量為 0，狗狗散步會增加身上的灰塵量，狗狗洗澡則會讓累積的灰塵量歸零。身高及體重是判斷狗狗是小型犬還是大型犬的依據，身高超過（不包含）60 公分或是體重超過（不包含）30 公斤的狗狗就是大型犬，反之則是小型犬。散步累積的灰塵量受狗狗的體型大小影響，小型犬累積的速度比較快，小型犬散步一次累積 3 單位的灰塵量，大型犬散步一次則累積 2 單位的灰塵量。

除了灰塵量，艾莎想要知道狗狗的散步狀況，可以利用兩項指標來衡量，分別為每一隻狗狗的散步頻率以及最大散步間隔時間。以下為兩項指標的計算方式說明：

- 散步頻率：將狗狗從被領養到今日所累積的散步次數除以從領養開始到現在這段期間累積的天數。
- 最大散步間隔時間：每次的散步間隔時間為從上一次到這一次散步的累積天數，每一次散步都可以計算得到此次散步的間隔時間，第一次散步的間隔時間，為從被領養的那天到第一次散步的累積天數，最後一次散步到今日的時間不納入考量。例如：領養日期為 2020 年 11 月 01 日，第一次散步為 2020 年 11 月 20 日，第二次散步為 2020 年 11 月 25 日，則第一次散步的間隔時間為 19 天，第二次散步的間隔時間為 5 天。在狗狗每一次散步的間隔時間中，最長的散步間隔時間即為此隻狗狗的最大散步間隔時間。已知每隻狗狗都至少散步過一次，且只考慮目前已有散步紀錄的間隔時間。

艾莎想要找出散步頻率最低的狗狗、最大散步間隔時間最長的狗狗以及灰塵累積量最多的狗狗，當出現多個狗狗的散步頻率都是最低的、多個狗狗最大散步間隔時間都是最長的或是多個狗狗的灰塵累積量都是最多的，篩選條件的優先順序是

1. 選擇大型狗狗
2. 選擇體重最重的狗狗
3. 選擇身高最高的狗狗
4. 名字開頭字母最前面的狗狗

你的程式要處理一連串的事件，一個事件用一行字串說明，這一行字串中會以半形逗點間隔數個資訊，已知狗狗們的名字不會重複，且狗狗第一次出現的事件必定為領養，一隻狗狗只會發生一次領養事件，在領養事件發生過後才会有此狗狗其他事件的紀錄，且若狗狗換主人後不會再出現這隻狗狗的任何事件紀錄，事件按照發生時間前後排序輸入。總共有四種事件，分別為艾莎領養狗狗、狗狗洗澡、狗狗散步以及狗狗換主人了，這些事件會影響目標狗狗的資訊。

這些關於狗狗的資訊該用什麼方式在程式中儲存起來呢？由於本週的課程就是介紹類別，在本題就讓我們來練習設計和實作類別吧！以本題的情境和任務來說，當然沒有非得要設計和實作類別不可，但就當作是教學目的，請大家還是乖乖跟著練習吧！

我們將在程式中建立名為 `Dog` 的類別，每個 `Dog` 物件裡面儲存該狗狗的名字、身高、體重、被領養的日期、累積灰塵量、散步次數、最大散步間隔時間、最近一次的散步日期以及是否為小型犬。你可以發現累積灰塵量在領養那天是從 0 開始累積的，所以在建立 `Dog` 這個類別的時候可以將累積灰塵量設定初始值為 0；同樣的道理也可以應用在散步次數，將散步次數的初始值設定為 0；最大散步間隔時間設定為 0；最近一次散步日期則以被領養日期作為初始值；是否為小型犬在建立物件的過程呼叫 `check_if_small_dog` 的 method 來計算此物件是否為小型犬，並以回傳值作為初始值。部分的程式碼如下：

```
class Dog:

    def __init__(self, name, height, weight, adopted_date):
        self.name = name
        self.height = height
        self.weight = weight
        self.adopted_date = adopted_date
        self.dust = 0
        self.walk_count = 0
        self.longest_duration = 0
        self.last_walk_date = adopted_date
        self.is_small_dog = self.check_if_small_dog()

    def check_if_small_dog(self):
        # 判斷是否為小型犬，回傳 boolean 值
        return 是否為小型犬

    def walk(self, walk_date):
        if self.is_small_dog:
            # 依據小型犬的灰塵累積效率更新累積灰塵量
        else:
            # 依據大型犬的灰塵累積效率更新累積灰塵量
        # 更新散步次數、最大散步間隔時間、最近散步日期

    def bathe(self):
        # 更新累積灰塵量
```

你的主程式應該依據一連串的事件，逐行讀取事件，每讀一列就對事件相關的狗狗進行計算，在處理完這些事件後也完成建立與更新這些物件。

一旦有了這許多物件，要完成以下四個任務就不會太難了。每項任務的輸出皆為一個字串包含四個部分，依序為目標狗狗的名字、身高、體重以及累積灰塵量，分別以半形逗號隔開。

- 任務 a：找出名字為給定名稱的狗狗。本題給定的名字不會出現艾莎沒有的狗狗或是已經換主人的狗狗。
- 任務 b：找出散步頻率最低的狗狗。本題不會出現沒有任一隻狗狗符合篩選條件的狀況。
- 任務 c：找出最大散步間隔時間最長的狗狗。本題不會出現沒有任一隻狗狗符合篩選條件的狀況。
- 任務 d：找出累積灰塵量最多的狗狗。本題不會出現沒有任一隻狗狗符合篩選條件的狀況。

請幫你的類別實作至少以下幾個 method（成員函數），分別為 `check_if_small_dog`、`walk` 以及 `bathe`。`check_if_small_dog` 會回傳此物件是否為小型犬，回傳值的型態為 `boolean`，依據此物件的身高體重計算出結果，在建構物件的時候使用。`walk` 沒有回傳值，此 method 依據散步日期更新此物件的累積灰塵量、散步次數、最大散步間隔時間以及最近一次的散步時間。`bathe` 沒有回傳值，用來更新累積灰塵量。若你覺得有幫助，你可以寫更多 method，但一定要有上述這三個。

舉例來說，如果事件的紀錄如表 1，狗狗的身高體重如表 2。以下模擬四種任務的狀況。

事件	狗狗的名字	事件發生日期
領養	Lucky	2020/01/01
散步	Lucky	2020/08/08
換主人	Lucky	2020/09/01
領養	Fatty	2020/10/10
散步	Fatty	2020/10/15
領養	Happy	2020/10/15
散步	Fatty	2020/10/20
散步	Happy	2020/10/20
洗澡	Fatty	2020/10/22
散步	Fatty	2020/10/24

表 1: 事件紀錄

狗狗的名字	身高	體重
Lucky	20	10
Fatty	35	15
Happy	65	25

表 2: 身高體重

假設在任務 a 的情境下，要找到名字為 Fatty 的狗狗資訊。Fatty 身高不超過 60 公分，體重不超過 30 公斤，是小型犬，單次散步累積灰塵量為 3 單位灰塵，在 10 月 22 以前累積的灰塵量為 6 單位，在洗澡之後灰塵累積量歸零，之後又散步一次，今日累積灰塵量為 3 單位。則輸出為 **Fatty,35,15,3**。

若執行任務 b 找出散步頻率最低的狗狗，則觀察每隻狗狗的散步頻率。Lucky 因為換主人了，所以不在考慮範圍；Fatty 從領養到今日共  $25 - 10 = 15$  天，累積三次散步，散步頻率為  $\frac{3}{15} = 0.2$ ；Happy 從領養到今日共  $25 - 16 = 9$  天，累積一次散步，散步頻率為  $\frac{1}{9} \approx 0.11$ 。散步頻率最低的狗狗是 Happy。Happy 的身高大於 60 公分，為大型犬，散步一次累積灰塵量為 2 單位，共散步一次，所以今日的累積灰塵量為 2，則輸出為 **Happy,65,25,2**

若執行任務 c 找出最大散步間隔時間最長的狗狗，Lucky 最大散步間隔時間為 220 天，但是 Lucky 換主人了，所以不在考慮範圍內；Fatty 最大散步間隔時間為 5 天；Happy 最大散步間隔時間為 5 天，因為 Happy 是大型犬，所以最大散步間隔時間最長的狗狗為 Happy，則輸出應該為 `Happy,65,25,2`。

若執行任務 d 找出累積灰塵量最多的狗狗，Lucky 不在考慮範圍內；Fatty 今日的灰塵累積量為 3 單位；Happy 的灰塵累積量為 2 單位，所以灰塵累積量最多的狗狗是 Fatty，則輸出為 `Fatty,35,15,3`。

## 輸入輸出格式

系統會提供一共數組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中，輸入資料共有  $n + 3$  行，第一行為今天的日期，格式為 `yyyy/mm/dd`，第二行為指定的任務類別，四種任務的輸入格式如下。

- 任務 a：輸入包含兩個部分，第一部分為一個字串「TaskA」，第二個部分是一個字串為狗狗的名字，這兩個部分以半形逗點間隔。
- 任務 b：輸入為一個字串「TaskB」。
- 任務 c：輸入為一個字串「TaskC」。
- 任務 d：輸入為一個字串「TaskD」。

第三行到第  $n + 2$  行為  $n$  個事件，直到第  $n + 3$  行輸入為「Done」，代表所有要進行的事件已經輸入完畢。所有事件的日期都不會晚於題目給定的今天日期。事件的輸入分為四種。

- 若要進行的事件為**愛莎領養狗狗**則輸入包含五個部分，第一個部分是一個字元「A」；第二個部分是一個英文字串，是這隻狗狗的名字；第三個部分是一個正整數，是狗狗的身高，第四個部分是一個正整數，是狗狗的體重，第五個部分是一個字串，代表領養的日期，格式為 `yyyy/mm/dd`，這五個部分中間以「|」隔開。
- 若要進行的事件為**狗狗洗澡**則輸入包含兩個部分，第一個部分是一個字元「B」；第二個部分是一個英文字串，是狗狗的名字，這兩個部分中間以「|」隔開。
- 若要進行的事件為**狗狗散步**則輸入包含三個部分，第一個部分是一個字元「W」；第二個部分是一個英文字串，是狗狗的名字；第三個部分是一個字串，代表散步的日期，格式為 `yyyy/mm/dd`，這三個部分中間以「|」隔開。
- 若要進行的事件為**狗狗換主人**則輸入包含兩個部分，第一個部分是一個字元「L」；第二個部分是一個英文字串，是狗狗的名字，這兩個部分中間以「|」隔開。

完成所有事件的運算後，依據給定任務找出目標狗狗，請印出一個字串，包含目標狗狗的名字、身高、體重以及累積灰塵量，分別以半形逗號隔開。

舉例來說，如果輸入是

```
2020/10/25
TaskA,Fatty
A|Fatty|35|15|2020/10/10
W|Fatty|2020/10/15
A|Happy|65|25|2020/10/16
```



```
W|Fatty|2020/10/20
W|Happy|2020/10/20
B|Fatty
W|Fatty|2020/10/24
Done
```

則輸出為

```
Fatty,35,15,3
```

如果輸入是

```
2020/10/25
TaskB
A|Fatty|35|15|2020/10/10
W|Fatty|2020/10/15
A|Happy|65|25|2020/10/16
W|Fatty|2020/10/20
W|Happy|2020/10/20
B|Fatty
W|Fatty|2020/10/24
Done
```

則輸出是

```
Happy,65,25,2
```

如果輸入是

```
2020/10/25
TaskC
A|Lucky|20|10|2020/01/01
W|Lucky|2020/08/08
L|Lucky
A|Fatty|35|15|2020/10/10
W|Fatty|2020/10/15
A|Happy|65|25|2020/10/15
W|Fatty|2020/10/20
W|Happy|2020/10/20
B|Fatty
W|Fatty|2020/10/24
Done
```

輸出應該是

```
Happy,65,25,2
```

如果輸入是

```
2020/10/25
TaskD
A|Fatty|35|15|2020/10/10
W|Fatty|2020/10/15
A|Happy|65|25|2020/10/15
W|Fatty|2020/10/20
W|Happy|2020/10/20
B|Fatty
W|Fatty|2020/10/24
Done
```

則輸出是

```
Fatty,35,15,3
```

## 你上傳的原始碼裡應該包含什麼

你的.py 原始碼檔案裡面應該包含讀取測試資料、做運算，以及輸出答案的 Python 3 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用上課沒有教過的方法。

## 評分原則

- 這一題的其中 20 分會根據程式運算的正確性給分。PDOGS 會直譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。本題共有 10 組測試資料，一筆測試資料佔 2 分，其中前 5 組只會出現任務 a，後 5 組則四種任務都有可能。
- 這一題的其中 10 分會根據你所寫的程式的品質來給分。助教會打開你的程式碼並檢閱你的程式的可讀性（包含排版、變數命名、註解等等）。請寫一個「好」的程式吧！除此之外，本題中你一定要實作那一個類別與三個 method，有缺的會被扣分，完全沒寫的整題的 30 分會被扣光。

## 第四題

（40 分）本題是一個 open question，也就是沒有標準答案的題目。因此，你不需要把針對本題寫出來的程式上傳到 PDOGS；你只需要利用你寫的程式產生一份報告，並且在 PDOGS 上上傳報告。

在本題中，你可以使用 csv 套件讀入給定的 submission\_complete.csv，這個檔案包含真實的本學期目前為止所有作業的繳交記錄，<sup>3</sup> 用任何你覺得合理、合適的方法去做資料分析與視覺化，「利用數據和證據」提供題目設計者檢視這些題目的難度與合適性。底下是幾個隨便舉的例子：

- 你或你身邊的同學，或許對某幾題感到特別困擾。針對那幾題，班上其他同學也有類似困擾嗎？他們有繳交特別多次嗎？

---

<sup>3</sup>本題中所有的 CSV 檔內的資料都可能包含中文，而所有中文都以 UTF-8 編碼，可以用例如 Notepad++ 等軟體開啟。

- 常常有同學反應作業太難，到底有多難？雖然繳交記錄不見得能反映同學們寫作業花的時間，但也可以反映一些事情，例如普遍來說比較困難的作業的平均每人繳交次數會高一點、平均 Accepted 繳交比率會低一點、Accepted 人數也會低一點。也許我們可以用這樣的邏輯找出比較難的題目？會不會其實大部分作業都還好，但就有幾份作業特別難？
- 常常有同學反應期中考週比較沒時間寫作業，我們能不能從資料中看看，是否確實有某些作業的繳交時間相對偏晚？「相對偏晚」是一個好指標嗎？要看平均數還是標準差？

你的報告應該要給出題者明確的建議，例如「作業四第三題太難了，因為如此這般，建議將來不要出這樣的題目」或「這些題目看似很難，但其實還好，因為每題每人平均只做了若干次繳交，且如此那般，建議以後將難度整體提升」。重點是，你的建議不能只是基於你或你身邊朋友的感覺，而是要基於你從整班所有人的所有繳交資料中看出的事實與證據，來提出你的建議。找出事實與證據是一回事，良好地呈現它們又是另一回事，此時製作一些圖表當然就很有幫助了。

針對本題，我們有以下幾個明確的規範：

1. 由於本週的課程介紹了 `matplotlib`，因此我們要強迫你使用這個函式庫：你的報告中一定要有至少三張用 `matplotlib` 畫出來的圖。
2. 你得要把你的報告做成一個 PDF 檔，最多四面，上傳到 PDOGS。由於上限是四面，顯然你的報告不能包山包海。請針對你的許多發現挑出最重要的幾個作呈現就好。
3. 請寫中文或英文，不要寫其他語言。
4. 在報告首面最上方應有你的系級、學號、姓名。

## 評分原則

- 只要你繳交一份滿足上述必要條件的報告，就能得到 20 分。
- 報告的其中 5 分會考慮你的報告是否排版合宜、行文流暢。
- 報告的其中 10 分會考慮你的圖表是否確實能支持你的建議（不論你的論點是否有價值）。
- 報告的最後 5 分會考慮你的建議的合理性與建設性。