

# 程式設計（109-1）

## 作業九

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/>) 為三題各上傳一份 C++ 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。請以英文或中文作答。

這份作業的截止時間是 **12 月 29 日早上八點**。在你開始前，請閱讀課本的第 22.1–22.6 節與第 9–10 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吳琦艾跟鄭亦辰。

### 第一題

（20 分）承作業八，本題一樣要進行撲克牌遊戲。場上共有四位玩家，每位玩家各有五張牌且皆不重複，請隨機將兩玩家組隊，並從十張牌中選出五張，目標依舊為最大化得分。舉例來說，若四位玩家的牌型分別為 {SA, S2, S3, H4, S5}、{SK, D7, CK, S4, S8}、{DJ, DQ, DK, DA, D2}、{HQ, HK, HA, H2, H3}，假設我們將玩家一和二組成一隊，即可從 {SA, S2, S3, H4, S5, SK, D7, CK, S4, S8} 中任選五張以最大化得分，以此例我們會選擇 {SA, S2, S3, S4, S5} 這五張牌，得分為 100。請依上述方式組合任意兩位玩家，並找出最高得分。

本題要求你實作 class 而非 structure，如無意外，應該會有 Card 和 Deck 這兩個 class。以下有幾點設計需要考慮。首先，請視情況將類別成員（變數或函數）設定為 `private` 或 `public`，原則是針對不希望隨意被更改或呼叫的變數或函數，請用 `private` 將它保護起來；第二個為 getter 和 setter 的設置，需自行權衡是否會與 `private` 的設置相衝突，你應該為有其需要的私有成員變數設置 getter 或 setter，但不應該幫每一個私有成員變數都把 getter 和 setter 全寫出來；第三個為 constructor 和 destructor 的設置，可自行設計並決定 constructor 的數量和其傳入參數的種類（例如是否要有 default constructor、是否要有帶參數的 constructor）、是否要有 destructor。

### 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。檔案的輸入格式和作業八第二題一模一樣。讀入輸入資料後，請根據上述規則，找出哪兩位玩家組隊後的五張牌可以得到最高分，並輸出一個正整數代表該分數。

舉例來說，如果輸入是

```
SA,S2,S3,H4,S5
SK,D7,CK,S4,S8
DJ,DQ,DK,DA,D2
HQ,HK,HA,H2,H3
```

則輸出應該是

---

<sup>1</sup>課本是 Deitel and Deitel 著的 *C++ How to Program: Late Objects Version* 第七版。

|     |
|-----|
| 100 |
|-----|

如果輸入是

|  |
|--|
| SA, S2, D2, S4, S5<br>SK, S7, S8, SQ, SJ<br>DJ, CQ, DK, DA, C2<br>HK, HA, H2, H9, C4 |
|--|

則輸出應該是

|    |
|----|
| 40 |
|----|

## 你上傳的原始碼裡應該包含什麼

你的.cpp 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 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用任何方法。

## 評分原則

這一題的 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。一筆測試資料佔 2 分。

## 第二題

(40 分) 現在我們要來玩一個撲克牌接龍遊戲，讓玩家把獲得的一疊撲克牌按照指定規則，盡可能地分成依花色區隔、依點數順序由小到大排好的四堆牌，然後計算被放入四個牌堆的總牌數。

規則大致如下。首先玩家會拿到一疊撲克牌，這疊牌的數量不固定，但確定沒有重複的牌。一開始桌面上無任何撲克牌，玩家每次翻開牌堆最上方的一張牌，就查看桌面上是否有可以接續此牌的牌堆，若有就放入那個牌堆接續，沒有就捨棄。由於四個花色會分開接續，因此一開始玩家應先以黑桃 A、紅心 A、方塊 A 和梅花 A 為目標，翻開的牌若非這四張，即可捨棄；若是其一，則可以放在桌面上，開始該花色的接龍，讓點數為二的同花色牌可以接在它後面，依此類推。最後整副撲克牌依上述規則翻開並各判斷一次後，我們會計算每種花色在桌面上的數量，並依黑桃、紅心、方塊、梅花的順序輸出該牌堆的牌數。

舉例來說，現有一副牌，依序分別為梅花 6、梅花 2、紅心 A、方塊 10、紅心 2、方塊 2、方塊 A、方塊 3，則玩家首先翻開第一張牌，發現梅花 6 無處可放（因為桌面上沒有梅花 5），則選擇丟棄；接著玩家翻開第二張梅花 2，由於梅花 2 也無處可放（因為桌面上沒有梅花 A），因此也丟棄；到了第三張，玩家翻開發現為紅心 A，便可以放在桌面作為「紅心」此花色的龍頭，換言之，現在玩家期待的牌型更新為黑桃 A、紅心 2、方塊 A 與梅花 A 了。按此規則，後面的方塊 10 和方塊 3 會被丟棄，而紅心 2 和方塊 A 會留在桌面。請注意雖然方塊 2 可以接在方塊 A 後面，但因為方塊 2 被翻開時方塊 A 還沒出現，因此方塊 2 只能被丟棄。結束後，計算各花色牌數，我們可以得到黑桃 0 張，紅心 2 張，方塊 1 張，梅花 0 張，因此輸出 0、2、1、0。

本題請利用前一題實做的 `class Card`，再實作一個 `class CardStack`，用來按照此題規則存放一疊撲克牌的花色和數字。在之前我們曾經實作過一個 `class Deck`，用來存放每位玩家的手牌，大家可以以此為參考，試著調整並設計出這次的 `class CardStack`。

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。每個檔案中會有一行，為一個字串，分別是第一張牌的花色、點數，第二張牌的花色、點數，直到第  $n$  張牌的花色、點數，兩兩中間以一個逗號隔開。每筆測資的  $n$  不固定， $1 \leq n \leq 52$ 。其中花色字元是 S、H、D、C 其中之一，點數字元則是 A、2、3、4、5、6、7、8、9、10、J、Q、K。 $n$  張牌不會重複。請根據上述規則，讀入一疊撲克牌後進行接龍遊戲，並計算各花色留在桌面上的數量，並依規則輸出四個整數，兩兩以一個逗號隔開。

舉例來說，如果輸入是

```
C6,C2,HA,D10,H2,D2,DA,D3
```

則輸出應該是

```
0,2,1,0
```

如果輸入是

```
S7,SA,CA,C2,C3,S6,S2,H5,C4,DA
```

則輸出應該是

```
2,0,1,4
```

## 你上傳的原始碼裡應該包含什麼

你的 .cpp 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 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**不可以**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 會編譯並執行你的程式、輸入測試資料，並檢查輸出的答案的正確性。一筆測試資料佔 2 分。
- 這一題的其中 20 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運算邏輯、可讀性，以及可擴充性（順便檢查你有沒有使用上課沒教過的語法，並且抓抓抄襲）。請寫一個「好」的程式吧！

除此之外，你**必須**實做題目指定的 `class`，並考慮相關設計，利用這些東西來完成運算。若你沒有這麼做，「程式的品質」部份將被扣分。

### 第三題

(40 分) 經濟學上有一個概念叫做帕累托效率 (pareto efficiency)，給定多個目標，不同的選項在各目標上會有不同的達成程度，而在帕累托前緣 (pareto frontier) 上的任何一點 (任何一種選項)<sup>2</sup>，代表沒有任何另一個選項會在各目標上同時優於這個選項。

舉例來說，假設現在我們要選擇一種交通方式前往某處，我們既希望時間盡量少，也希望花費盡量少，而這些選項中包括了腳踏車、計程車、公車與走路，每種選項皆有各自的花費時間和成本。我們估算了各種選項的所需時間與成本，並在二維平面上表示這些選項，如圖 1 所示。我們可以看到，對於計程車、腳踏車和走路這三種選項的任意一種，並沒有任何另一種選項同時在時間和成本花費上少於它，因此藍色的線條代表帕累托前緣，而這三種選項都在此前緣上；反之，由於腳踏車這個選項比公車這個選項花費時間和成本皆較少 (可能剛好這個人要去的地方搭公車不太方便)，所以公車並不在帕累托前緣上。

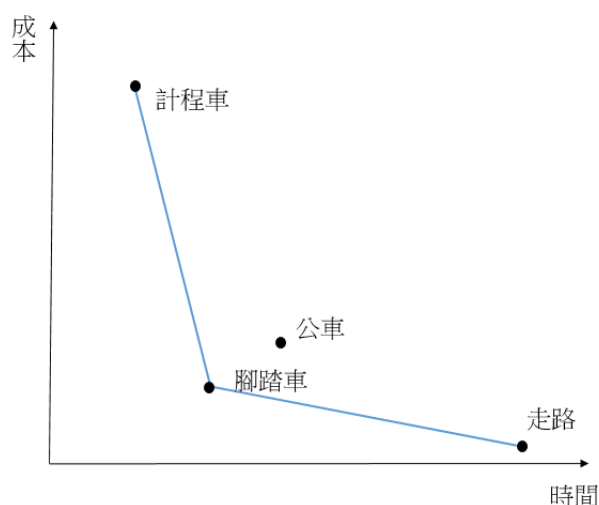


圖 1: 帕累托前緣舉例示意圖 1

本題會給定  $n$  個候選選項  $x_1$  到  $x_n$ ，以及  $m$  個目標，各目標數值皆為越小越好。第  $i$  個選項在  $m$  個目標上的數值分別為  $x_{i,1}$ 、 $x_{i,2}$ ，直到  $x_{i,m}$ 。給定兩個選項  $x_i$  和  $x_j$ ，若  $x_i$  在各目標上表現皆優於或等於  $x_j$  在各目標上的表現，且至少有其中一個目標  $x_i$  表現優於  $x_j$ ，也就是

$$x_{it} \leq x_{jt} \quad \forall t = 1, 2, \dots, m,$$

且存在至少一個  $t_0$  使得

$$x_{i,t_0} < x_{j,t_0},$$

則我們說  $x_i$  「優於」 (dominate)  $x_j$ 。給定  $n$  個選項，請實作一個 `class Option`，去記錄各選項的編號以及在各目標上的數值，以及一個成員函式

```
bool Option::isDominating(Option b)
```

傳入另一個選項，判斷呼叫此函數的選項是否優於傳入的選項，以及另一個函式

<sup>2</sup>更精確的定義在下一段提供。

```
bool Option::isBefore(Option b)
```

傳入另一個選項，判斷呼叫此函數的編號是否小於傳入的選項的編號。請利用這個 class 找出所有在帕累托前緣上的選項，並根據選項之編號由小到大依序輸出各選項在各目標上的數字。

舉例來說，若給定  $n = 4$ ， $m = 2$ ，選項一到四在兩目標上的數字分別為  $(2, 2)$ 、 $(1, 5)$ 、 $(4, 3)$ 、 $(5, 1)$ （在二維平面上圖示如圖 2），則在帕累托前緣上的選項為編號 1、2 和 4，因此依序輸出  $(2, 2)$ 、 $(1, 5)$ 、 $(5, 1)$ 。

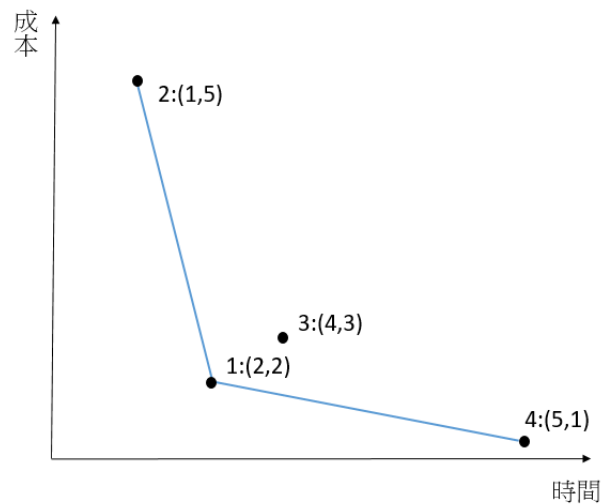


圖 2: 帕累托前緣舉例示意圖 2

## 輸入輸出格式

系統會提供數組測試資料，每組測試資料裝在一個檔案裡。每個檔案中會有  $n + 1$  行，第一行有兩個數字  $n$  和  $m$ ，分別代表選項數量和目標數量，其中  $1 \leq n \leq 100$ ， $1 \leq m \leq 10$ 。第二到第  $n + 1$  行各有  $m + 1$  個數字，在第  $i + 1$  行中，第一個數字為該選項編號  $i$ ，第二到  $m + 1$  的第  $t + 1$  個數字為該選項在各目標上的數字  $x_{it}$ ，其中  $0 \leq x_{it} \leq 1000$ ，各數字間兩兩以一個空白隔開。請根據上述規則，找出在帕累托前緣上的選項，並依照編號由小到大，依序輸出這些選項在各目標式的值，目標式之間以一個逗點隔開，選項之間以一個分號隔開。

舉例來說，如果輸入是

```
4 2
4 5 1
1 2 2
3 4 3
2 1 5
```

則輸出應該是

```
2,2;1,5;5,1
```

如果輸入是

```
5 4
1 2 3 1 2
4 0 5 0 1
3 0 0 0 2
2 7 9 8 1
5 10 12 9 7
```

則輸出應該是

```
0,0,0,2;0,5,0,1
```

## 你上傳的原始碼裡應該包含什麼

你的.cpp 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 C++ 程式碼。當然，你應該寫適當的註解。針對這個題目，你**可以**使用任何方法。

## 評分原則

這一題的 40 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。一筆測試資料佔 2 分。