

# 資料結構與進階程式設計 (107-2)

## 作業三

作業設計：孔令傑

國立臺灣大學資訊管理學系

繳交作業時，請將第一題的答案以中文或英文作答後，以 PDF 檔上傳到 NTU COOL；**不接受紙本繳交**；第二至四題請至 PDOGS (<http://pdogs.ntu.im/judge/>) 上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）。這份作業的截止時間是 **2019 年 4 月 8 日凌晨 1:00**。在你開始前，請閱讀課本的第 3、4 章<sup>1</sup> 不接受遲交。

由於本課程修課同學眾多，而助教人力有限，為了避免過度壓榨助教，我們可能會對手寫題進行隨機批改，亦即助教會隨機選擇一份作業的若干題進行批改和給分。助教會對全班同學都批改同樣的題目。若有批改之題目的總分為  $x$  而一位同學得到  $y$  分，且手寫題總分為  $z$  分，則其在該份作業會得到  $z(\frac{y}{x})$  分。公佈的參考解答中會包含所有題目（包含沒被批改的）。

## 第一題

（40 分；每小題 10 分）請回答以下各題：

(a) 請到 Wikipedia 看 Catalan number 的說明<sup>2</sup>，並寫一個能以遞迴方式計算 Catalan number 的函數，此函數的參數為一個整數  $n$ ，回傳第  $n$  個 Catalan number。

(b) 我們教過「給定  $n$  個物品，重量各為  $w_1, w_2$  到  $w_n$ ，請找到其中數個東西，能恰好塞滿負重量上限為  $B$  的包包，或判定不可能」，這樣的一個 knapsack problem（更精確地說，一個 decision-version 的 knapsack problem），有 pseudo-polynomial 的演算法。

現在，請考慮「給定  $n$  個物品，重量各為  $w_1, w_2$  到  $w_n$ ，價值各為  $v_1, v_2$  到  $v_n$ ，包包負重量上限為  $B$ ，請找到其中數個東西，能不違反負重限制並且讓總價值達到  $V$  以上，或判定不可能」的問題。請也設計一個 pseudo-polynomial 的演算法，去解決這個問題。請注意你並不需要讓所挑選的東西的重量總和恰好為  $B$ ，只要小於等於  $B$  就好了。

(c) 針對本次作業第二題（在下面）指定的 augmenting path algorithm 實做方式，請幫它寫下 pseudocode。

(d)（每子題 5 分）你正在處理一個物流問題，是給定一個有向圖  $G = (V, E)$ 、每個點代表一個運送路徑上的地點（工廠、倉庫、物流中心、市場等等）、每個邊代表兩個地點間的一條路、邊  $(i, j)$  上有最大可運送的數量  $u_{ij}$ ，而你要從指定的起點  $s \in V$  運送盡量多貨品到指定的終點  $t \in V$ 。毫無疑問地，這是一個 maximum flow problem。

你在本次作業的第三題，寫了一個可以解 maximum flow problem 的程式。高興之餘，你的老闆又給了你新的物流問題。你感覺它們都很像 maximum flow problem，但又不太一樣。

(i) 給定  $G = (V, E)$ ，你要從三個起點  $v_1, v_2$  和  $v_3$  出發，運送盡量多的貨品到兩個終點  $t_1$  和  $t_2$ 。從任意一個指定起點運送任何數量的貨品到任意一個指定終點，該數量都可以被加進你的總運送量。

---

<sup>1</sup>課本是 Carrano and Henry 著的 *Data Abstraction and Problem Solving with C++: Walls and Mirrors* 第六版。

<sup>2</sup>[https://en.wikipedia.org/wiki/Catalan\\_number](https://en.wikipedia.org/wiki/Catalan_number)。

- (ii) 給定  $G = (V, E)$ ，現在不只是路段上有流量上限，連地點也有流量上限。更精確地說，若地點  $i \in V$  的流量上限為  $U_i$ ，則流經該地點的流量總和（總共送進去該地點的量，或總共從該地點送出來的量）不能超過  $U_i$ 。

針對以上兩個情境，請說明如何利用你已經寫好的 maximum flow 程式，去求得最佳的運送方案。

## 第二題

（30 分）在本題中，我們將請你實做上課教的 augmenting path algorithm，不過你不用完成整個實做，只要照規則找出一條 augmenting path 就好了。本題要求你使用上課教的 BFS（breath-first search，廣度優先）方法，在每一階層中考慮由  $s$  通往目標節點之子路徑的最大運量，由大而小放進 search queue（搜尋佇列）中，若平手則將最終節點編號較小的先放入 queue。請在找到第一條可以由  $s$  運送到  $t$  的路徑時結束，並且輸出該路徑。

讓我們以圖 1 為例。根據上課教的 BFS 搜尋法，我們將執行以下步驟：

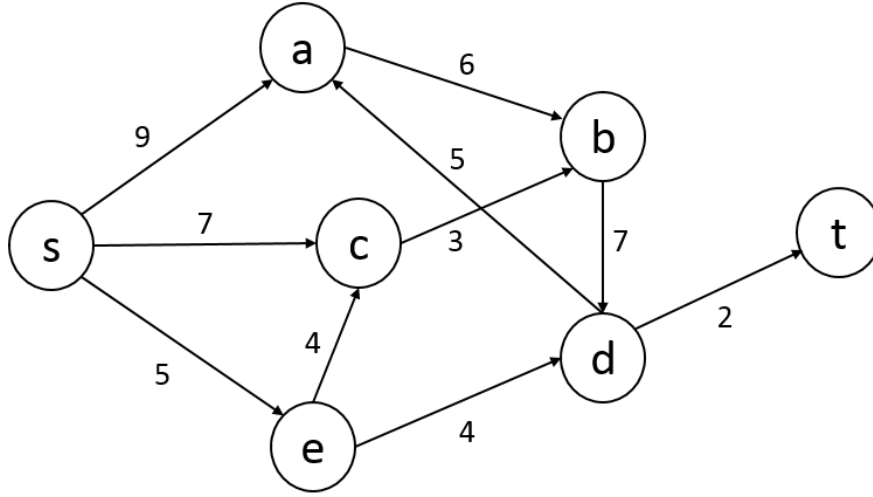


圖 1: Augmenting path 搜尋範例

1. 首先我們要在 queue 中放入  $(s, a; 9)$ 、 $(s, c; 7)$  與  $(s, e; 5)$  三個路徑，由於  $9 > 7 > 5$ ，我們將按照上面的順序放入 queue。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, a; 9), (s, c; 7), (s, e; 5)\}。$$

2. (a) 接著我們從 queue 中拿出  $(s, a; 9)$ ，放入從  $a$  再走一步的  $(s, a, b; 6)$ 。請注意由  $a$  往下只有一個節點  $b$ ，且由於  $u_{ab} < 9$ ，放入 queue 中的路徑的最大運量只有 6。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, c; 7), (s, e; 5) \mid (s, a, b; 6)\}。$$

那一槓之前的是步驟一放入的路徑（長度為 1），之後的則是步驟二放入的路徑（長度為 2）。

- (b) 接著我們從 queue 中拿出  $(s, c; 7)$ ，放入從  $c$  再走一步的  $(s, c, b; 3)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, e; 5) \mid (s, a, b; 6), (s, c, b; 3)\}。$$

- (c) 接著我們從 queue 中拿出  $(s, e; 5)$ ，放入從  $e$  再走一步的  $(s, e, c; 4)$  與  $(s, e, d; 4)$ 。由於兩個路徑的最大運量相同，我們將最終節點編號較小的先放入 queue。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, a, b; 6), (s, c, b; 3), (s, e, c; 4), (s, e, d; 4)\}。$$

3. (a) 接著我們從 queue 中拿出  $(s, a, b; 6)$ ，放入從  $b$  再走一步的  $(s, a, b, d; 6)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, c, b; 3), (s, e, c; 4), (s, e, d; 4) \mid (s, a, b, d; 6)\}。$$

- (b) 接著我們從 queue 中拿出  $(s, c, b; 3)$ ，放入從  $b$  再走一步的  $(s, c, b, d; 3)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, e, c; 4), (s, e, d; 4) \mid (s, a, b, d; 6), (s, c, b, d; 3)\}。$$

- (c) 接著我們從 queue 中拿出  $(s, e, c; 4)$ ，放入從  $c$  再走一步的  $(s, e, c, b; 3)$ 。完成此步驟後，queue 中含有的路徑依序為

$$\{(s, e, d; 4) \mid (s, a, b, d; 6), (s, c, b, d; 3), (s, e, c, b; 3)\}。$$

- (d) 接著我們從 queue 中拿出  $(s, e, d; 4)$ ，考慮從  $d$  再走一步的  $(s, e, d, a; 4)$  與  $(s, e, d, t; 2)$ 。此時由於已經發現從  $s$  到  $t$  的一個路段了，因此結束搜尋，並且印出該路段。

## 輸入輸出格式

系統會提供一共 15 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  $m + 2$  列資料，其中第一列是兩個正整數  $n$  和  $m$ ，第二列是  $n$  個只含有最多十個大寫英文字母的字串  $s_1, s_2$  直到  $s_n$ ，其中  $s_i$  代表節點  $i$  的編號（本題中我們用字串當編號），且  $s_1$  為運送起點、 $s_n$  為運送終點。第三列起的每一列含有兩個字串  $x_j, y_j$  與一個非負整數  $u_{x_j, y_j}$ ，代表有向路段  $(x_j, y_j)$  存在於給定的 network 中，且其運量上限為  $u_{x_j, y_j}$ 。每一列中相鄰的兩個值都以一個逗點隔開。已知  $1 \leq n \leq 20$ 、 $1 \leq m \leq n(n - 1)$ 、 $0 \leq u_{ij} \leq 100$ ，給定的路段資訊不包含重複的路段，但可能包含兩個節點間兩個方向（一來一往）的資訊。

讀入以上資訊後，請按照題目指定的方法找出一條 augmenting path，並依序印出該路徑由起點  $s_1$  開始要通過的節點，一直到終點  $s_n$  為止，最後印出該路徑的最大運量。任兩個印出的節點編號之間以一個逗點隔開，終點  $s_n$  與最大運量之間用一個分號隔開。如果給定的 network 中不存在 augmenting path，就輸出  $-1$ 。舉例來說，若輸入為

```
7,10
s,a,b,c,d,e,t
s,a,9
s,c,7
s,e,5
d,t,2
d,a,5
b,d,7
e,c,4
```

```
e,d,4  
c,b,3  
a,b,6
```

則輸出應該為

```
s,e,d,t;2
```

若輸入為

```
7,10  
s,a,b,c,d,e,t  
s,a,9  
s,c,7  
s,e,5  
d,t,0  
d,a,5  
b,d,7  
e,c,4  
e,d,4  
c,b,3  
a,b,6
```

則輸出應該為

```
-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

這一題的分數都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。

## 第三題

(30 分) 承上題，現在我們要請你在找到 augmenting path 之後把它用掉、修改 residual network，然後反覆執行直到找到最佳解（最大流量）為止。本題的輸入方式和第二題一模一樣，輸出則改為每找到一條 augmenting path，就用上一題的方式在新的一行輸出它，持續下去直到那一輪找不到 augmenting path，此時則在新的一行輸出一個整數，為給定的 network 的最大流量。注意這個數字也有可能是 0。舉例來說，若輸入為若輸入為

```
7,11
s,a,b,c,d,e,t
s,a,9
s,c,7
s,e,5
d,t,2
d,a,5
b,d,7
b,t,7
e,c,4
e,d,4
c,b,3
a,b,6
```

則輸出應該為

```
s,a,b,t;6
s,c,b,t;1
s,e,d,t;2
9
```

## 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

這一題的分數都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。

## 第四題（加分題）

（20 分）承作業二第二題，排班演算法運作模式依舊，在此題有修改的部分只在於演算法的第一步驟，船舶工作的排序規則。令船舶  $j$  的可出發時間為  $r_j$ （把 00:00 當成 0、00:01 當成 1、01:00 當成 60，依此類推）、總作業時間（若為進或出港則為兩段作業時間的合計）為  $p_j$ ；若它是危險船舶（即有裝載危險貨品）則  $d_j = 1$ ，反之  $d_j = 0$ ；此外，本題還會給定一個優先進出國清單資訊，該檔案裡面記載所有優先進出國的國名，若船舶  $j$  是屬於優先進出國，則  $q_j = 1$ ，反之則  $q_j = 0$ 。則船舶工作的「排序規則」計算方式如下：

1. 根據給定的係數  $\beta_1$ 、 $\beta_2$ 、 $\beta_3$  與  $\beta_4$ ，每條船根據上述變數定義與下述公式計算一個分數

$$s_j = b_1 r_j + b_2 p_j - b_3 d_j - b_4 q_j。$$

2. 考慮計算出來之分數  $s_j$ ，分數愈小者愈優先安排；若分數平手則比危險指標  $d_j$ ，危險者優先安排；若再平手則比「優先進出國指標」 $q_j$ ，屬於優先進出國者優先安排；若再平手情況下，則比船舶編號，字典順序小者優先安排。

優先進出國清單檔案中，國名記載規則如下。若一國國名只有一個字，則記全名，例如 TAIWAN 就會記 TAIWAN、CHINA 就會記 CHINA。若國名超過一個字，則記每一個字的地一個字母合併的縮寫，例如 UNITED STATES 就記 US。這個檔案中每行記載一個國家名稱，沒有非大寫英文字母的符號。沒有任兩個國家在優先進出國清單中的記載方式會一模一樣。

## 輸入輸出格式

系統會提供一共 10 組測試資料，每組測試資料裝在一個檔案裡。在每個檔案中總共會有四行，第一行是給定的排班參數，共有五個整數依序為  $m$ 、 $\beta_1$ 、 $\beta_2$ 、 $\beta_3$ 、 $\beta_4$ ，任兩個數字之間用一個逗點隔開。第二行會有一個檔案路徑的字串，表示船舶資訊檔案的存放位置。第三行會有一個檔案路徑的字串，表示工作資訊檔案的存放位置，第四行會有一個檔案路徑的字串，表示優先進出國清單資訊檔案的存放位置。已知  $0 \leq \beta_i \leq 1000$ 、 $2 \leq m \leq 200$  且  $m$  為偶數。在依序讀入這些資訊後，請分別印出四個數字，依序是所有一港進港船的延遲時間、所有一港出港船的延遲時間、所有二港進港船的延遲時間，以及所有二港出港船的延遲時間，兩個數字中間以一個空白字元隔開。

船舶資訊與工作資訊檔案內容記載形式皆如同前一題，優先進出國清單資訊檔案中則將依據前所述的記載規則記錄屬於優先進出的國家。舉例來說，若 AnotherShipSample.txt 為船舶資訊檔案，其內容為

```
AFECXS,25500,TAIWAN,LING KUNG,Y
AFEUTI,20000,JAPAN,HIRONA OISHI,N
TRUCXS,13700,THAILAND,SAWADIKA,Y
THUEDC,18600,TAIWAN,YENYENER,N
KCUHIC,78600,THAILAND,KAPUNKA,Y
ITCUOW,12000,TAIWAN,JASON WANG,Y
ZCHCBE,88888,JAPAN,KANAKO,N
```

而 AnotherWorkSample.txt 為工作資訊檔案，其內容為

```
AFECXS,10:00,30,20,I,1,12
AFEUTI,10:40,12,T,106,120
TRUCXS,10:10,10,5,0,1,15
THUEDC,10:10,10,10,0,1,35
KCUHIC,10:20,25,15,I,1,150
ZCHCBE,13:00,10,30,0,2,35
```

最後，PrioritySample.txt 為優先進出國清單資訊檔案，其內容為

```
TAIWAN
JAPAN
US
PHILIPPINE
```

```
KOREA
CANADA
```

則綜合上述，針對本題題意，如果輸入是

```
200,5,70,8,7
AnotherShipSample.txt
AnotherWorkSample.txt
PrioritySample.txt
```

則最後輸出應該是

```
75 5 0 0
```

## 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

這一題的分數都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。