

資料結構與進階程式設計 (108-2)

作業六

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/>) 為第一題上傳一份 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。

這份作業的截止時間是 **2020 年 4 月 21 日早上八點**。在你開始前，請閱讀課本的第 2、5 和 10 章¹。為這份作業設計測試資料並且提供解答的助教是楊其恆。

第一題

(20 分) 第一題我們將實作 Knapsack 問題的 optimization 版本，也就是在課程投影片第 41 頁的題目。在本題中，請實作最簡單的貪婪演算法，也就是在一堆物品中，每次選擇 C/P 值最大的 (也就是 v_i 除以 w_i 最大的) Item，放入背包中，直到沒有任何物品可以再被放到背包中為止，若有多個 C/P 值同為最小的物品則請先將編號較小的放入背包中。要注意的是，若你發現某個物品沒有辦法放進背包中，仍然有可能會有 C/P 值比他小的物品可以被放進背包中，所以不能一發現有一個物品放不進去背包就停止搜尋。

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。每個檔案會有 3 行，第一行包含兩個整數 n 和 B ，分別代表總物品數量以及背包重量上限；第二行有 n 個整數，依序是 v_1, v_2 直到 v_n ，表示這 n 個物品的價值；第三行有 n 個整數，依序是 w_1, w_2 直到 w_n ，表示這 n 個物品的重量。已知 $1 \leq n \leq 100, 1 \leq B \leq 10000, 1 \leq v_i \leq 100, 1 \leq w_i \leq 100$ 。

讀入上述資訊之後，請根據題意計算並輸出所有被選到背包中的物品的編號以及背包的總重量，背包內的物品請以編號由小到大排序，每兩個數字間以一個逗號隔開，物品編號及背包總重量間則以一個分號隔開。舉例來說，若輸入為

```
8 20
10 13 8 15 24 12 10 20
3 7 5 2 6 5 10 5
```

則輸出應該為

```
1,4,5,8;16
```

在此範例中，物品被選入背包的順序依序為 4、5、8、1，但輸出時依照編號大小排序。

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

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，你可以使用任何語法。

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) 第二題我們將實作 decision 版的 Knapsack 問題，也就是我們要找出是否存在一個物品集合，使該集中所有物品重量的總和剛好等於背包耐重上限。

在投影片的第 45 到第 49 頁，老師有介紹 Recursive 和 Dynamic Programming 這兩種方式可以解這個問題。在本題的測試資料中，會有幾筆非常大的資料是無法在時間限制內用 Recursive 的方式解出來的，只有使用 Dynamic Programming 才能在時間限制中解出來，因此請你使用 Dynamic programming 的方式實作。

如果存在有物品集合可以恰好使重量總和等於背包耐重上限，則可能會有多組解。此時請使用投影片所教的方式，從真正要解的子問題（在投影片鐘用 $P(B, n)$ 表示）往前回溯，當遇到一個物品可選也可以不選時，請選擇之（換言之，在 NS 和 S 之間以 S 為主）。以投影片第 48 的範例為例，如果 $B = 5$ ，因為 $P(5, 4)$ 可為 S 或 NS，我們應該選 S，因次應該輸出的物品集合只包含物品 4，而不是包含物品 1 和 2。

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。每個檔案會有 2 行，第一行包含兩個整數 n 和 B ，分別代表總物品數量以及背包重量上限；第二行有 n 個整數，依序是 w_1 、 w_2 直到 w_n ，表示第 i 個物品的重量。已知 $1 \leq n \leq 1000$ 、 $1 \leq B \leq 10000$ 、 $1 \leq w_i \leq 100$ 。

讀入上述資訊之後，請根據題意計算並輸出所有被選到背包中的物品之編號，背包內的物品以編號由小到大排序，兩個編號之間以一個逗點隔開，若不存在一個物品集合的總和剛好等於背包總和，請輸出 0。

舉例來說，若輸入為

4 16
8 3 4 5

則輸出應該為

1,2,4

若輸入為

```
4 16
8 3 4 6
```

則輸出應該為

```
0
```

若輸入為

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

則輸出應該為

```
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語法。

評分原則

- 這一題的其中 30 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 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！

第三題

（40 分）在本題中，我們將求解 bipartite matching 問題，而我們的求解方式是去解 maximum flow 問題的一個特例：一個所有 edge 的 capacity 都是 1 的 bipartite graph。底下我們將說明為什麼解掉這樣的 maximum flow 問題就可以解出 bipartite matching，而求解 maximum flow 的演算法請參考投影片第 54 到 63 頁。

首先讓我們來介紹 bipartite matching。假設有三個人與三個工作，其中員工 1 有能力做工作 1 和工作 2、員工 2 有能力做工作 1 和工作 3、員工 3 也是有能力做工作 1 和工作 3。假設一個員工只能做一件工作，我們該如何分配工作給員工，盡可能完成最多工作呢？

我們可以將這個員工與工作的關係抽象化，畫成如圖 1 所示的一個 bipartite graph。在這個圖中，左邊那一排 node 代表員工，由上往下依序是員工 1、2、3，右邊那一排 node 代表工作，由上往下依序是工作 1、2、3，若兩點間有連線則表示該員工能做該工作。

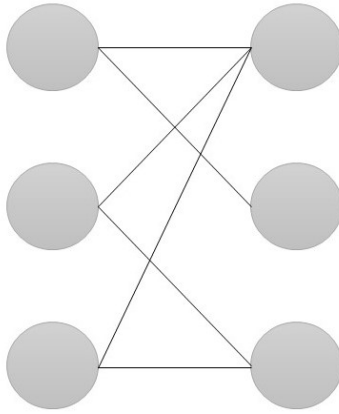


圖 1: Bipartite matching 範例圖

在一個 graph 上，一組匹配 (matching) 是一個 edge 的集合，其中任兩條 edge 都沒有在同一個點上交會，此時一組 matching 裡面的任意一個 edge 就被稱為一個 match，而 edge 的個數就被稱為匹配數。如果我們想要最大化被完成的工作數，就在這個 bipartite graph 中最大化匹配數就好了。

bipartite matching 有很多解法，而在此題中，我們將藉由解延伸出的 maximum flow 來解此問題。我們首先把一個 bipartite graph 上的 edge 都改成 directed，從左邊那一排連向右邊那一排，然後加上兩個 node s 和 t ，讓 s 連到所有的左邊那一排 node，讓 t 被所有的右邊那一排 node 連到。以圖 1 為例，這樣做完之後會得到如圖 2 的 graph。我們再將此 graph 中所有的 capacity 都設為 1，這樣就定義好一個 maximum flow 問題了。如果我們能實作 augmenting path algorithm，就能找出圖 2 中的最大流量，那我們就會馬上得到圖 1 中的最大匹配了²。

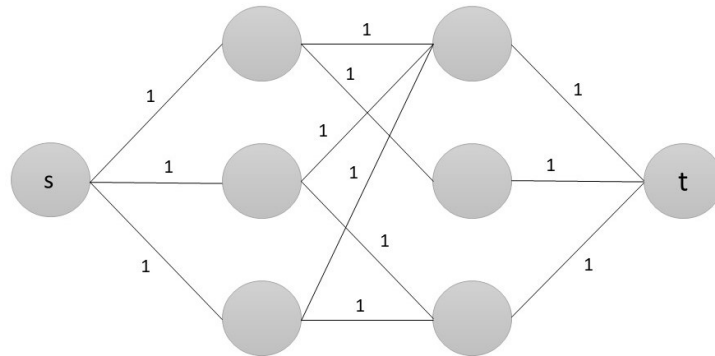


圖 2: Maximum flow 範例圖

雖然也可以用其他方法來解 bipartite matching 的問題，但因為在作業七中，我們將實作 general 的 augmenting path algorithm，所以我們強烈建議你在這次作業就練習實作 augmenting path algorithm。但你應該也有發現，在 general 的情況下實作 augmenting path algorithm 會需要做 graph search，去找出一條從 s 走到 t 的 path，這件事通常都需要使用到 queue 跟 stack 的概念，現在就要求大家實作是稍嫌太早了。這是為什麼這次我們只讓大家考慮 bipartite matching 延伸而出的 maximum flow，在圖 2 這樣的塗上要找從 s 到 t 的 path 當然是容易多了。

²雖然這需要一點嚴謹的證明，但我們就不折磨各位同學了。

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。每個檔案會有 2 行，第一行包含一個整數 n ，表示在此 bipartite graph 的一排中有幾個點。令 m_i 為左邊那一排的第 i 個點的 degree（連結數）， N_{ij} 為第 i 左邊的點的第 j 個在右邊那排的 neighbor 點的編號，檔案的第二行即包含 N_{ij} 的資訊，依序為 N_{11} 、 N_{12} ，直到 N_{1,m_1} ，兩兩之間以一個逗點隔開；接著是一個分號，接著是 N_{21} 、 N_{22} 直到 N_{2,m_2} ，兩兩之間依然以一個逗點隔開，以此類推到 N_{n,m_n} 。以圖 1 為例，其輸入會是

```
3
1,2;1,3;1,3
```

已知 $1 \leq n \leq 10000$ 、 $1 \leq N_{ij} \leq n$ 、 $1 \leq j \leq m_i$ 。

讀入上述資訊之後，請根據題意計算並輸出在此 bipartite graph 的最大配對中總共有幾個配對。舉例來說，若輸入為

```
3
1,2;1,3;1,3
```

則輸出應該為

```
3
```

若輸入為

```
3
1;1,3;1
```

則輸出應該為

```
2
```

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語法。

評分原則

- 這一題的其中 30 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 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！