程式設計（106-1）
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、二題上傳一個 PDF 檔，再為第三題上傳一份 C++ 原始碼（以複製貼上原始碼的方式上傳）。第四題是 bonus 加分題。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 ***2017 年 10 月 31 日凌晨一點***。在你開始前，請閱讀課本的第 5.1–5.7 和 6.5–6.8 節[1]。為這份作業設計測試資料並且提供解答的助教是楊佩蓉。

第一題

（20 分）請先看完第三題和第四題的題目，再來做這一題：

(a)（5 分）請說明在第三題範例中給的函數在那些陣列參數前面加上 `const` 會有什麼效果，以及為什麼要這麼做。

(b)（5 分）承上題，為什麼在那些非陣列參數前面不加 `const`？

第二題

（20 分）請先看完第三題和第四題的題目，再來做這一題：

(a)（10 分）承上題，你的朋友跟你說，不如把 `ss`、`ele`、`costs`、`cover` 宣告成四個 global 陣列，這樣就不用一直要傳陣列進去函數，不然每次都傳一大堆數字進去函數，很浪費時間。請問這種說法有什麼錯誤？

(b)（10 分）承上題，你的另一個朋友跟你說了剛剛那位朋友的論點何處有誤，但他還是建議你宣告成四個 global 陣列，並且表示這樣執行起來多少還是有比較快，程式碼也比較精簡。他這樣說固然是沒錯（而且 global 陣列有其他好處），但這樣做還是有壞處。請說明有什麼壞處。

提示：寫成 global 陣列的話，還能宣告為 `const` 嗎？此外，會破壞模組化嗎？

第三題

（60 分）有一個 Computer Science 領域的經典問題叫做「set cover」，大意如下：給定一個集合、一個集合裡面的許多元素，以及許多子集合，如何選取盡量少的子集合來包含到所有元素？一個現實的例子是蓋基地臺：給定全臺灣所有有住人的地點、所有可以蓋基地臺的地點，以及每一個基地臺可以覆蓋哪些地點的資訊後，我們想要蓋基地臺去覆蓋所有有住人的地點，並且蓋愈少愈好。在稍微比較複雜一點的版本中，每個子集合各有一個成本，我們想要讓總成本盡可能地低，而不是讓選取的子集合數量盡量

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

少。以基地臺的例子來說，每個地點蓋基地臺的成本可能各不相同，因此目標設定為花最少的錢可能比蓋最少基地臺更合理。

Set cover 是一個經典的 NP-hard 問題，只要題目規模夠大（例如有幾千個元素、幾千個子集合），在合理的時間內我們（目前）就求不出最佳解。在本題中，我們將給你一個 set cover 問題，並且請你使用以下的演算法求出一個解。這個演算法是一個迭代式（iterative）的啟發性演算法（heuristic algorithm），在每一次迭代（iteration）中，我們會對每一個還沒被選的子集合 i 計算「若選了它，能再包含 n_i 個還沒被包含的元素」以及選它要花的成本 c_i，然後在所有還沒被選的子集合中選 $\frac{c_i}{n_i}$ 最小的（也就是 CP 值最高的）。如果平手，則選編號最小的子集合（如果編號是字元，就按照字典順序）；如果一個子集合不能包含任何未包含的元素，當然就不會被選。我們持續這麼做直到包含所有元素為止。請注意在考慮一個子集合的效益時，我們關心的是在當下選它可以再包含多少還沒被包含的元素。這個值會根據已選的子集合而持續變化。

我們用一個例子來說明這個題目和演算法。在圖 1 中，我們有五個基地臺（在五個大圈圈的中心點）A、B、C、D 和 E，各需要 10、8、6、4、7 元。每個基地臺各可以覆蓋到一些小鎮，例如 A 可以覆蓋到編號 1、2、5、6 的小鎮、B 可以覆蓋到編號 3、5、6、9 的小鎮，依此類推。我們的目標是花最少的錢蓋基地臺，來覆蓋到所有小鎮。若我們用 set cover 的語言來說，則我們說我們有一個集合 $S = \{1, 2, ..., 9\}$，內含 9 個元素，以及 5 個子集合 $A = \{1, 2, 5, 6\}$、$B = \{3, 5, 6, 9\}$、$C = \{1, 2, 7, 8\}$、$D = \{4\}$、$E = \{2, 3, 4, 5\}$，各需要成本 $c_A = 10$、$c_B = 8$、$c_C = 6$、$c_D = 4$、$c_E = 7$。我們要花盡量少的成本來選擇子集合以包含到所有的元素。

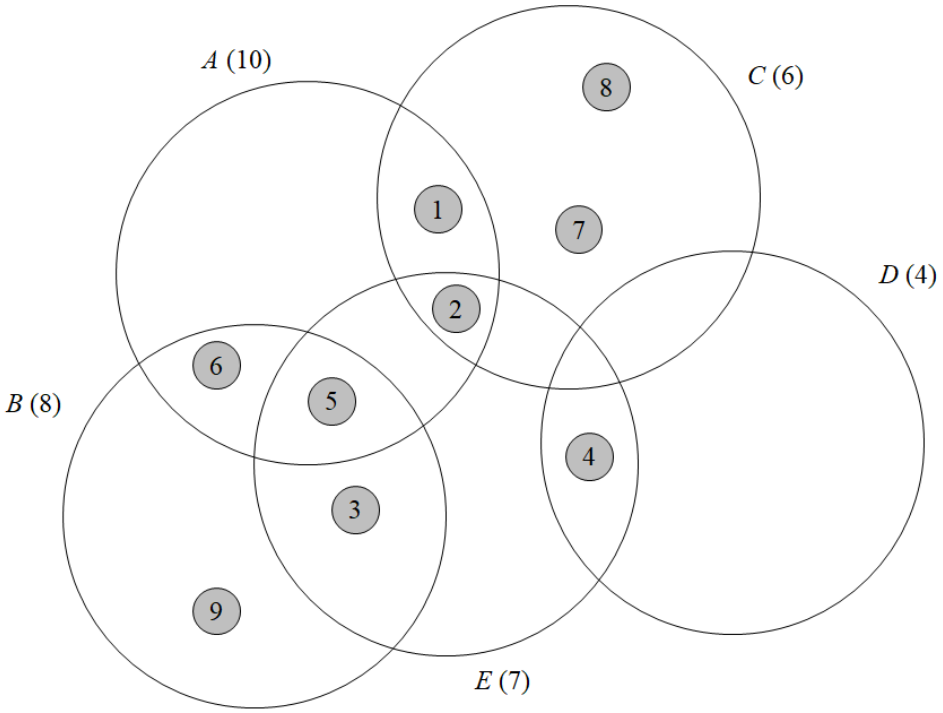

圖 1: Set cover 的例子

給定這個題目以及我們的演算法，我們會依照以下順序選擇子集合：

1. 在第一輪的選擇中，對於五個未被選取的子集合 A、B、C、D 和 E，他們的 $\frac{c_i}{n_i}$ 各是 $\frac{10}{4}$、$\frac{8}{4}$、$\frac{6}{4}$、$\frac{4}{1}$、$\frac{7}{4}$。因為子集合 C 的 $\frac{c_i}{n_i}$ 最小，也就是成本效益最大，我們選 C。

2. 在第二輪的選擇中，對於四個未被選取的子集合 A、B、D 和 E，他們的 $\frac{c_i}{n_i}$ 各是 $\frac{10}{2}$、$\frac{8}{4}$、$\frac{4}{1}$、$\frac{7}{3}$。因為子集合 B 的 $\frac{c_i}{n_i}$ 最小，也就是成本效益最大，我們選 B。

3. 在第三輪的選擇中，對於四個未被選取的子集合 A、D 和 E，他們的 $\frac{c_i}{n_i}$ 各是 $\frac{10}{0}$、$\frac{4}{1}$、$\frac{7}{1}$。因為子集合 D 的 $\frac{c_i}{n_i}$ 最小，也就是成本效益最大，我們選 D。

由於已經包含所有元素了，演算法至此結束，依序選了三個子集合 C、B、D，共花了 18 元。

在本題中，我們將請你用上述演算法求解給定的 set cover 問題。你或許已經發現這個問題的每一輪都在做很像的事情，而一輪裡面對每一個未被選擇的子集合，我們也都做很像的事情，因此都很適合寫 function 並且傳入不同 argument 以得到結果。以後者為例，假設我們最多有 100 個元素，我們可以寫一個 function

```
float score(int ssIndex, const bool ss[], const bool ele[],
            const int costs[], const bool cover[][100]),
            int ssCnt, int eleCnt);
```

其中 `ssIndex` 是一個我們想計算其成本效益的子集合（subset，縮寫為 ss）的陣列 index、`ss` 記錄哪些子集合已經被選擇了、`ele` 記錄哪些元素（element，縮寫為 ele）已經被選擇了、`costs` 記錄各子集合的成本、`cover` 記錄一個子集合是否包含一個元素、`ssCnt` 記錄共有幾個子集合、`eleCnt` 記錄共有幾個元素。我們讓這個函數回傳在目前被選取的狀態下，若選擇 `ssIndex` 這個子集合會得到多大的成本效益，則在一輪內我們只要反覆呼叫這個函數（並且在 `ssID` 傳入不同內容），就可以得到所有的成本效益，接著我們就可以知道應該選誰了。請注意：

- 我們傳進去的是子集合的索引值，不是子集合的編號，因為這樣才知道怎麼使用其他那些陣列。
- 你不一定要寫上述那樣的函數，你可以寫你想要的函數。舉例來說，你傳入的 `cover` 可以不用是二維陣列（記錄全部的包含關係），而是只傳入一個一維陣列（你要計算的子集合的包含關係）；你也可以寫一個直接算出應該選哪個子集合的函數（那當然參數組合也會不同），它可以用到上面給的那個函數（或者也可以不要用）。總之，你得練習自行決定了。
- 那些矩陣參數都被設定為 `const` 了。為什麼？

當然，你甚至可以不寫函數就完成這一題，但若是有寫合理的函數，一定會讓這個程式比較容易理解，也會節省你的開發時間（因為思路會比較清晰）。愈早開始練習寫函數，當然也會讓你的程式設計能力愈好，也愈能挑戰更困難、更複雜的程式。

輸入輸出格式

系統會提供一共 20 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 $n+3$ 行，第一行包含兩個整數 n、m，分別是子集合的個數和元素的個數；第二行至第 $n+1$ 行的第 $i+1$ 行包含 k_i+1 個整數，其中第一個數字 k_i 表示第 i 個子集合能包含哪些元素，後面 k_i 個數字不重複，代表子集合 i 能包含的元素的編號；第 $n+2$ 行包含 n 個不重複的英文大寫字元，第 i 個字元是子集合 i 的編號；第 $n+3$ 行包含 n 個整數，第 i 個整數是子集合 i 的成本 c_i。元素編號為 1、2、3 直到 m。同一行的任意兩個整數之間被一個空白隔開。已知 $1 \le n \le 25$、$1 \le m \le 100$、$1 \le k_i \le m$、$1 \le c_i \le 1000$，並且每一個元素都被至少一個子集合包含。

讀入資料後，請按照題目指定的規則與執行順序，依序印出被選擇的子集合的編號。任兩個編號之間用一個空白字元隔開。舉例來說，如果輸入是

```
5 9
4 1 2 5 6
4 6 5 3 9
4 8 7 1 2
1 4
4 2 5 3 4
A B C D E
10 8 6 4 7
```

則輸出應該是

```
C B D
```

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 40 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！

第四題

（20 分）承上題，我們想要稍微修改一下我們的演算法。仔細觀察圖 1 的例子，我們可以發現子集合 B 是非選不可的，不然無從包含元素 9。我們可以在一開始先選擇這些非選不可的子集合（按照字母順序），再開始執行我們在第三題指定的演算法[2]。以圖 1 為例，我們選擇子集合的順序會變成 B、C、D，雖然最後選出來的子集合和第三題一樣，但順序和第三題不一樣。

本題的輸入和輸出格式都和第三題一模一樣，只是演算法不同。舉例來說，如果輸入跟第三題的範例一樣，則輸出應該是

```
B C D
```

針對這個題目，你***可以***使用任何方法。這一題的 20 分都根據程式運算的正確性給分，一筆測試資料佔 2 分。

[2]這樣做有什麼好處？