

程式設計 (106-1)

作業七

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、四題各上傳一份 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 11 月 7 日凌晨一點**。在你開始前，請閱讀課本的第 5.20-5.22 和第 19 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

(10 分) 請看 https://en.wikipedia.org/wiki/Catalan_number，去學習關於 Catalan numbers 的相關知識。請寫一個 C++ 函數

```
int catalan(int n);
```

在給定 $n \geq 0$ 之後，用遞迴的方式找出第 n 個 Catalan number C_n 。舉例來說， $C_0 = 1$ 、 $C_1 = 1$ 、 $C_2 = 2$ 、 $C_3 = 5$ 。請不要在意 overflow 的事情 (就是假設 n 不會太大的意思)。

第二題

(30 分) 請回答以下關於 sorting 的問題：

- (a) (10 分) 請用 insertion sort 去把 (7, 4, 6, 3, 8, 2, 9, 0, 1, 5) 由小到大排序，並且寫下每一輪 insert 完之後的陣列內容 (第一輪排完第一個數字，第 k 輪排完前 k 個數字)。舉例來說，如果用 insertion sort 排 (4, 3, 2, 5)，答案應該是

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

- (b) (10 分) 請到 https://en.wikipedia.org/wiki/Selection_sort 或其他網頁，去瞭解「selection sort」的作法。用自己的話簡單說明之後，去把 (7, 4, 6, 3, 8, 2, 9, 0, 1, 5) 由小到大排序，並且寫下每一輪結束之後的陣列內容 (第一輪排完第一個數字，第 k 輪排完前 k 個數字)。舉例來說，如果用 selection sort 排 (5, 4, 2, 3)，答案應該是

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

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

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

- (c) (10 分) 請到 https://en.wikipedia.org/wiki/Bubble_sort 或其他網頁，去瞭解「bubble sort」的作法。用自己的話簡單說明之後，去把 (7, 4, 6, 3, 8, 2, 9, 0, 1, 5) 由小到大排序，並且寫下每一輪結束之後的陣列內容（第一輪排完第一個數字，第 k 輪排完前 k 個數字）。舉例來說，如果用 bubble sort 排 (5, 4, 2, 3)，答案應該是

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

第三題

(60 分) 在本題中，我們將延續作業六的 set cover 問題，去考慮一個更接近現實的情況。在實務上，有時我們在蓋一些設施的時候，可以選擇要不要多花一點錢來蓋出威力更強的設施，例如能覆蓋更大面積的基地臺、能照顧到更多人的醫院之類的。因此在一個 set cover 問題中，當我們要挑選一個子集合時，我們可能可以決定要讓這個子集合達到何種規模 (scale)，較大的規模可以包含更多元素，但也要花更多成本。我們的目標依然是最小化包含所有元素所需的成本。

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（在五個實線大圈圈的中心點） 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。其中對於基地臺 C 我們可以選擇兩種規模，小規模需要 6 元，只能覆蓋實線圈圈內的四個小鎮，大規則則需要 9 元，但可以覆蓋虛線圈圈內的五個小鎮。

若我們用 set cover 的語言來說，則我們說我們有一個集合 $S = \{1, 2, \dots, 9\}$ ，內含 9 個元素，以及 5 個子集合 $A = \{1, 2, 5, 6\}$ 、 $B = \{3, 5, 6, 9\}$ 、 $C = \{1, 2, 7, 8\}$ 、 $D = \{4\}$ 、 $E = \{2, 3, 4, 5\}$ ，其中子集合 C 有兩種規模 $\{1, 2\}$ 。六個子集合與規模的組合各需要成本 $c_A = 10$ 、 $c_B = 8$ 、 $c_{C_1} = 6$ 、 $c_{C_2} = 9$ 、 $c_D = 4$ 、 $c_E = 7$ 。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。請注意這個問題並不同於一個有六個子集合可以選的問題，因為子集合 C 是不能選兩次的。

在本題中，我們將請你實做和作業五的第三題很像的演算法。在這個演算法中，每一次迭代我們會對每一個還沒被選的子集合 i 的規模 j 計算「若在此子集合選到此規模，能再包含 n_{ij} 個還沒被包含的元素」以及選它到該規模要花的成本 c_{ij} ，然後在所有還沒被選的子集合規模組合中選 $\frac{c_{ij}}{n_{ij}}$ 最小的（也就是 CP 值最高的）。子集合內的數個規模有同樣效益時，選規模編號最小的；數個子集合的最高效益規模有同樣效益時，選編號最小的子集合（如果編號是字元，就按照字典順序）。如果一個子集合的規模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。

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：

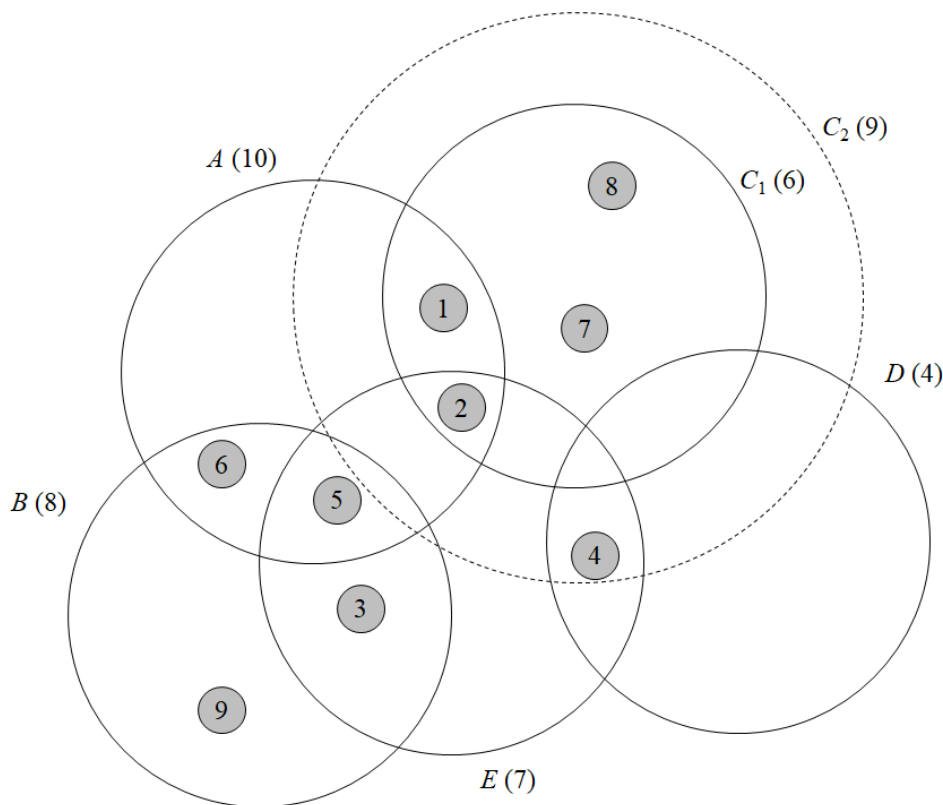


圖 1: Set cover 的例子

1. 在第一輪的選擇中，對於五個未被選取子集合 A 、 B 、 C 、 D 和 E ，他們的 $\frac{c_{ij}}{n_{ij}}$ 各是

$$\frac{c_{A,1}}{n_{A,1}} = \frac{10}{4}, \frac{c_{B,1}}{n_{B,1}} = \frac{8}{4}, \frac{c_{C,1}}{n_{C,1}} = \frac{6}{4}, \frac{c_{C,2}}{n_{C,2}} = \frac{9}{5}, \frac{c_{D,1}}{n_{D,1}} = \frac{4}{1}, \frac{c_{E,1}}{n_{E,1}} = \frac{7}{4}.$$

因為子集合 C 規模 1 的 $\frac{c_{ij}}{n_{ij}}$ 最小，也就是成本效益最大，我們選 C 的第一個規模。

2. 在第二輪的選擇中，對於四個未被選取子集合 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 。請注意此時我們不能考慮 C 的第二個規模，因為 C 已經被決定要蓋第一個規模了。
3. 在第三輪的選擇中，對於四個未被選取子集合 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。顯然若我們選 C 的第二個規模和 B ，也能包含所有元素，而且只要 17 元，但給定的演算法並沒有發現這件事。

在本題中，我們將請你用上述演算法求解給定的 set cover 問題。為了簡單起見，我們假設每個子集合最多只有兩種規模。

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。在每個檔案中會有若干行，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。第二行包含 n 個整數，依序表示第 i 個子集合

的規模數。第三行起的每一行表示一個子集合規模組合的相關資訊。首先，第一個值會是一個介於 A 到 Z 之間的大寫英文字元，表示子集合編號；第二個值是一個整數 1 或 2，表示這是第幾個規模；第三個值是一個整數 c_{ij} ，代表選取該子集合到此規模的成本；第四個值是一個整數 k_{ij} ，表示此子集合的子規模能包含 k_{ij} 元素；後面 k_{ij} 個數字不重複，代表子集合 i 的規模 j 能包含的元素的編號（1、2、3 直到 m ）。同一行的任意兩個值之間被一個空白隔開。已知 $1 \leq n \leq 25$ 、 $1 \leq m \leq 100$ 、 $1 \leq k_{ij} \leq m$ 、 $1 \leq c_{ij} \leq 1000$ ，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規模組合的編號。印出一個子集合規模組合的方式是先印出子集合編號，接著印出一個半形破折號（-），接著印出規模編號。如果某子集合只有一個規模，就不要印出半形破折號和規模編號。任兩個組合之間用一個空白字元隔開。舉例來說，如果輸入是

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

則輸出應該是

```
C-1 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（bonus）

（20 分）在本題中，我們將請你實做河內塔的程式。給定 n 個圓盤，由小到大編號為 1、2 到 n ，一開始放在柱子 A，要利用柱子 B 去被移動到柱子 C，請先找出最快速的圓盤移動方案，接著印出在此方

案中第 i 步應該把編號幾號的圓盤從哪個柱子移到另一個柱子。舉例來說，假設給定 4 個圓盤，則最佳方案是如下的 15 步：

1. Disc 1 from A to B
2. Disc 2 from A to C
3. Disc 1 from B to C
4. Disc 3 from A to B
5. Disc 1 from C to A
6. Disc 2 from C to B
7. Disc 1 from A to B
8. Disc 4 from A to C
9. Disc 1 from B to C
10. Disc 2 from B to A
11. Disc 1 from C to A
12. Disc 3 from B to C
13. Disc 1 from A to B
14. Disc 2 from A to C
15. Disc 1 from B to C

如果題目要求你印出第 6 步，則請印出 `Disc 2 from C to B`；如果題目要求你印出第 11 步，則請印出 `Disc 1 from C to A`；

如果 n 很大，在題目允許的時間內你恐怕無法列出所有的步驟，但當然你也不需要，因為列到第 i 步也就夠了。若你願意，你甚至可以經過精密的數學計算去直接得到第 i 步的內容（舉例來說，當有 4 個圓盤時，第 8 步顯然就是 `Disc 4 From A to C`）。總之，請試試看吧！

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。在每個檔案中會有一行，依序記錄 n 和 i 。讀入資料後，請按照題目指定的方式，印出第 i 步的內容。已知 $1 \leq n \leq 30$ 、 $1 \leq i \leq 2^n - 1$ 。舉例來說，如果輸入是

4 6

則輸出應該是

<code>Disc 2 from C to B</code>

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。