

中原大學資訊工程系 演算法分析第一次機測

Deadline: 4 / 24 / 2020 (星期五)
(限期中考前一週測完，逾期不得補繳)

【程式設計說明】

1. 每組限 2~3 人，組員須固定，本學期不得任意變更。原則上以專題組員為主。
2. 組員應合作共同解題，但嚴禁跨組合作。
3. 程式設計必須使用 Python 程式語言，版本採用 3.7 (原則上下載與安裝 Anaconda)。
4. 可參考課本、相關書籍或 Algorithms.py 等解題，解題方法及演算法不限，但絕對嚴禁抄襲他組程式，組員均有責任保護程式不被他組抄襲。若發現抄襲屬實，兩組均以零分計。
5. 所有輸入及輸出均為標準格式，即程式在命令提示字元環境下執行時可以鍵盤輸入資料，本機測不採讀檔方式進行。
6. 每一支程式均須附上組員姓名及學號，例如：

```
# 演算法分析機測  
# 學號: 10427XXX / 10427XXX  
# 姓名: 江○○ / 李○○  
# 中原大學資訊工程系
```

程式命名依該組學號在前之同學 [學號+題號] 為原則。例如：

```
10427001_1.py  
10427001_2.py
```

【機測須知】

1. 評分以解題成功之題數多寡與執行時間決定。
2. 程式必須能處理不同之輸入資料 (但輸入格式與範例相同)，並輸出正確結果 (輸出格式必須與範例相同)，組員應能說明程式設計內容，方可視為成功。程式之輸出結果錯誤、輸出格式與範例不符、或在執行後超過 60 秒仍未結束，均視為失敗。若程式測試失敗給予基本分數，未繳交程式則以零分計。
3. 本機測於規定之期限前，各組應攜帶程式原始碼至電學大樓 603 室找助教測試 (電話：265-4726)，每組限繳交一次，不可分題或多版本繳交，逾期不得補繳。
4. 助教將使用不同之輸入資料作為測試與評分依據，同學應在繳交前充分測試程式。
5. 機測成績納入學期平時成績計算，請同學把握！

指導教授: 張元翔

I. 最大子陣列問題 (Maximum-Subarray Problem)

最大子陣列問題 (Maximum-Subarray Problem) 在電腦演算法中是一個相當重要的問題，問題描述如下：給定一整數陣列 (Array)，目的在找到子陣列 (Subarray)，即連續整數和，且其總和最大。

試根據課本之 Divide-and-Conquer 演算法實現 C/C++ 程式用來解最大子陣列問題。

輸入說明：

輸入包含幾組資料，每組資料以一正整數 n 開頭，代表輸入之整數個數，若為 0 代表結束。接著為 n 個整數，每個整數以空格隔開。

輸出說明：

根據每組資料輸出最大子陣列的最小索引、最大索引及最大總和。

輸入範例：

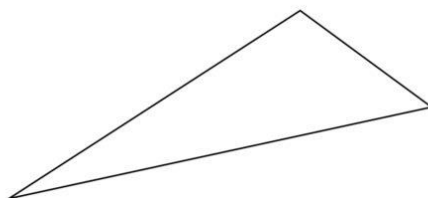
```
8
-2, 1, -3, 4, -1, 2, 1, -5
16
13 -3 -25 20 -3 -16 -23 18 20 -7 12 -5 -22 15 -4 7
0
```

輸出範例：

```
Low = 4, High = 7, Sum = 6
Low = 8, High = 11, Sum = 43
```

II. 三角形個數

假設給定 n 個長度不同的桿子，長度分別為 $1、2、\dots、n$ ，我們從中選出三個不同的桿子組成一個三角形，請問共有幾種不同的組合方式？



輸入說明：

每組測試資料會有一個整數 n ， $3 \leq n \leq 100$ ，若為 0 代表結束。

輸出說明：

輸出組合的方法有幾種。

輸入範例：

5
8
0

輸出範例：

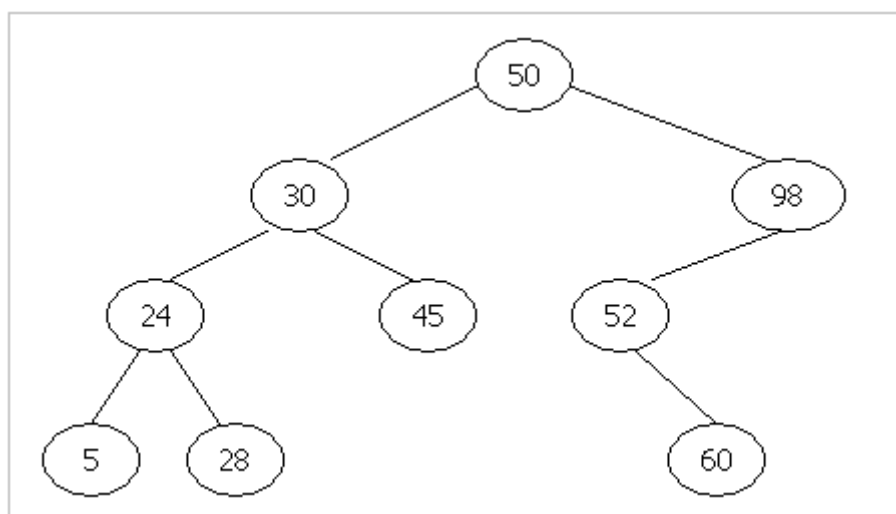
3
22

III. 二元搜尋樹

二元搜尋樹 (Binary Search Tree) 是一種二元樹且滿足以下性質：

- 某節點左子樹的節點的值都小於該節點的值。
- 某節點右子樹的節點的值都大於該節點的值。
- 左子樹和右子樹也必須為二元搜尋樹。

以下為二元搜尋樹的範例：



要拜訪一棵二元樹所有的節點有幾種不同的方式。**前序追蹤** (Preorder traversal) 會列印根節點的值，然後拜訪並列印左子樹，最後拜訪並列印右子樹。而**後序追蹤** (Postorder traversal) 則先拜訪並列印左子樹，再拜訪並列印右子樹，最後才列印根節點的值。上圖兩種方式列印節點值的順序如下：

Preorder: 50 30 24 5 28 45 98 52 60

Postorder: 5 28 24 45 30 60 52 98 50

若給你一棵二元搜尋樹**前序追蹤**的結果，請輸出其**後序追蹤**的結果。注意：若為二元搜尋樹，則答案為唯一。

輸入說明：

輸入為二元搜尋樹前序追蹤的結果，輸入值可能為正整數或大寫英文字母 A、B、...等，並以空格隔開。0 則代表結束。

輸出說明：

輸出二元搜尋樹後序追蹤的結果，每顆二元搜尋樹以一系列為原則，節點值以空格隔開。

輸入範例：

50 30 24 5 28 45 98 52 60

E C A B D G F

0

輸出範例：

5 28 24 45 30 60 52 98 50

B A D C F G E

IV. 數獨遊戲

數獨是一種數學邏輯遊戲，遊戲是由 9×9 個格子組成，玩家須根據格子提供的數字推理出其他格子的數字。遊戲設計者提供至少 17 個數字使得解答謎題只有一個答案。數獨的答案必須滿足：無論是哪一列、哪一行、或是九宮格內，1~9 的數字都僅出現一次。

5	3			7					5	3	4	6	7	8	9	1	2
6			1	9	5				6	7	2	1	9	5	3	4	8
	9	8					6		1	9	8	3	4	2	5	6	7
8				6				3	8	5	9	7	6	1	4	2	3
4			8		3			1	4	2	6	8	5	3	7	9	1
7				2				6	7	1	3	9	2	4	8	5	6
	6					2	8		9	6	1	5	3	7	2	8	4
			4	1	9			5	2	8	7	4	1	9	6	3	5
				8			7	9	3	4	5	2	8	6	1	7	9

現在請您使用程式設計，用來玩數獨遊戲，並提供答案。

輸入說明：

輸入 9×9 個格子，其中包含 0~9，0 代表空格。

輸出說明：

輸出 9×9 個格子為答案。

輸入範例：

530070000

600195000

098000060

800060003

400803001

700020006

060000280

000419005

000080079

輸出範例：

534678912

672195348

198342567

859761423

426853791

713924856

961537284

287419635

345286179

V. 方塊拼圖 (Grid Puzzle)

下圖為一典型的方塊拼圖 (Grid Puzzle)，說明如下：

0	5	8
7	4	3
2	6	1

方塊拼圖之構成：

1. 大小固定為 3×3 。
2. 0 固定出現在左上角，1 固定出現在右下角。
3. 其他數字介於 2~8 之間。

方塊拼圖之移動：

1. 0 代表空格，因此可以將緊鄰 0 之數字搬到空格中。
2. 搬動的成本 (Cost) 剛好等於所搬動的數字本身。

舉例而言：

0	5	8
7	4	3
2	6	1

搬動 5 到 0 的成本為 5

搬動 7 到 0 的成本為 7

但不可搬動其他數字到 0

方塊拼圖之目標：

1. 將右下角之 1 搬到左上角，因此搬動的方法有很多種，但搬動的成本總和不同。
2. 找到搬動成本總和最小者並列印結果。

舉例說明：

0	5	8
7	4	3
2	6	1

Step 1

成本: 5

5	0	8
7	4	3
2	6	1

Step 2

成本: 8

5	8	0
7	4	3
2	6	1

Step 3

成本: 3

5	8	3
7	4	0
2	6	1

Step 4

成本: 1

5	8	3
7	4	1
2	6	0

成本總和 = $5 + 8 + 3 + 1 + \dots$

輸入說明：

輸入為 3×3 之方塊拼圖。

輸出說明：

輸出最小成本總和。

輸入範例：

0 5 8

7 4 3

2 6 1

輸出範例:

Minimum Sum of Costs = 46