

中原大學資訊工程系 演算法分析第一次機測

Deadline: 4 / 23 / 2021 (星期五)
(限期中考當週測完，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 3.8 (請下載與安裝 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: 10727XXX / 10727XXX  
# 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：

```
10727001_1.py  
10727001_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，描述如下：給定一整數陣列 (Array)，其中可能包含正或負整數，目的是找到子陣列 (Subarray)，即連續元素整數和，且其總和最大。請參考課本 (講義) 之 Divide-and-Conquer 演算法，試設計程式解最大子陣列問題。

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 則代表結束。接著為 n 個整數，每個整數以空格隔開。你可以假設所有整數均介於 $-100 \sim 100$ 之間。

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. 撲克牌 24 點遊戲

(Poker 24-Point Game)

現有一副撲克牌，每張牌的數值均介於 1~13 之間（即 A 視為 1，J、Q、K 分別視為 11、12、13，本問題不考慮花色）。撲克牌 24 點遊戲的玩法如下：

- 從這副撲克牌中任意抽出 4 張牌給玩家（數值有可能重複）。
- 玩家可將這 4 張牌的數值任意排列，然後採用加、減、乘、除四則運算（即+、-、*、/等），運算時允許小數存在，並且可以使用括號。但是，每張牌只可以使用一次。
- 玩家嘗試建構一運算式，使其運算結果為 24。

請你根據上述遊戲規則，試設計程式幫忙玩家解決問題。

輸入說明：

每列表示任意抽出的一組撲克牌，分別為 4 張牌的數值（中間以空格隔開），0 0 0 0 則代表結束。

輸出說明：

若能得到運算結果為 24，則輸出相對應的運算式，否則輸出無解 (No Solution)。

輸入範例：

11 8 3 5
3 3 7 7
9 9 6 2
1 1 1 2
0 0 0 0

輸出範例：

$(11-8)*(3+5)=24$
 $7*(3+3/7)=24$
 $9*(2+6/9)=24$
No Solution

注意：運算應依先乘除後加減， $7*(3+3/7)$ 若輸出為 $7*(3+(3/7))$ 仍視為是不正確的輸出。

III. 簡易微分方程式解題器

(Simple Differential Equation Solver)

翔哥在中原大學資訊工程系教授工程數學多年，現在需要一套簡易的微分方程式解題器，除了可以幫助教學外，也可以驗證答案是否正確。身為軟體工程師，試設計程式協助翔哥完成這一套簡易的微分方程式解題器。

這個簡易微分方程式解題器主要須能針對高階微分方程式 ($n=2$ 或 3) 自動求解，其中僅包含常係數微分方程式，且均為齊次方程式。例如：

(1) $y'' + 3y' + 2y = 0$ 的解為： $y = c_1 e^{-x} + c_2 e^{-2x}$

(2) $y'' - 4y' + 4y = 0$ 的解為： $y = c_1 e^{2x} + c_2 x e^{2x}$

(3) $y'' + 3y = 0$ 的解為： $y = c_1 \cos(\sqrt{3}x) + c_2 \sin(\sqrt{3}x)$

(4) $y'' + 2y' + 17y = 0$ 的解為： $y = e^{-x}(c_1 \cos(4x) + c_2 \sin(4x))$

(5) $y''' + 3y'' - 4y = 0$ 的解為： $y = c_1 e^x + c_2 e^{-2x} + c_3 x e^{-2x}$

注意微分方程式的自變數為 x ，應變數為 y 。微分均使用微分運算子的型態表示之。例如：
 y'' 以 D^2y 表示； y' 以 Dy 表示等。此外，若解包含根號，以 $()^{1/2}$ 表示。

輸入說明：

輸入包含幾個微分方程式，0 代表結束。

輸出說明：

列出微分方程式的解；若微分方程式不在定義範圍內，則列出 Not Supported!

輸入範例：

$D^2y + 3Dy + 2y = 0$

$D^2y - 4Dy + 4y = 0$

$D^2y + 3y = 0$

$D^2y + 2Dy + 17y = 0$

$D^3y + 3D^2y - 4y = 0$

$$Dy+3*y=0$$

$$D^2y+3*Dy+2*y=\exp(x)$$

0

輸出範例:

$$y=c_1*\exp(-x)+c_2*\exp(-2*x)$$

$$y=c_1*\exp(2*x)+c_2*x*\exp(2*x)$$

$$y=c_1*\exp(3^{1/2}*x)+c_2*\sin(3^{1/2}*x)$$

$$y=\exp(-x)*(c_1*\cos(4*x)+c_2*\sin(4*x))$$

$$y=c_1*\exp(x)+c_2*\exp(-2*x)+c_3*x*\exp(-2*x)$$

Not Supported!

Not Supported!

IV. B 樹 (B-Trees)

在資訊科學中，B-Trees 是一個相當特別的資料結構，假設其階數 (Order) 為 m ，則 B-Trees 可定義如下：

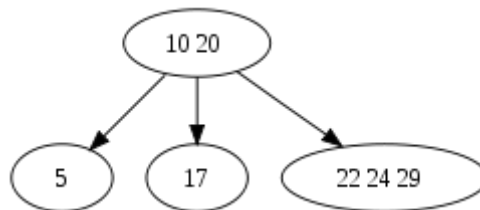
- (a) 每一個節點含有 $\lceil m/2 \rceil \sim m$ 子節點
- (b) B-tree 為一平衡樹，亦即所有的樹葉節點之高度均相同

以階數 $m = 4$ (即 2-3-4 Tree) 為例，則每個節點可能是 2-Node (含 1 個 Key 及 2 個子節點)、3-Node (含 2 個 Keys 及 3 個子節點)、4-Node (含 3 個 Keys 及 4 個子節點)。

若要在 2-3-4 Tree 插入某 Key 值，則操作均從樹根開始搜尋：

- (1) 若搜尋過程中，遇到的節點為 4-Node，需先做 Split，否則繼續往樹葉節點搜尋。
- (2) 搜尋至樹葉節點後，在該節點插入 Key 值。

以下範例而言：



則插入 25 後可得到 2-3-4 Tree 如下 (其中發生 Split):



以本範例而言，其前序及後序走訪分別為：

Preorder: (10, 20, 24) (5) (17) (22) (25, 29)

Postorder: (5) (17) (22) (25, 29) (10, 20, 24)

試實現 2-3-4 Tree 的插入 (Insertion) 操作。

輸入說明

給定一組正整數 (介於 1~99) 之間作為 Key 值，分別以空格隔開，依序插入及建立 2-3-4 Tree。

輸出說明

輸出其前序及後序走訪，每個節點均以括號為之，Key 值以逗點隔開。

輸入範例

20 50 40 70 80 15 90 100

輸出範例

2-3-4 Tree (Preorder):

(50) (30) (10, 15, 20) (40) (70) (60) (80, 90, 100)

2-3-4 Tree (Postorder):

(10, 15, 20) (40) (30) (60) (80, 90, 100) (70) (50)

V. 水桶謎題

假設有兩個水桶及一個水池 (無限供應水)，兩個水桶的容量均為已知，但是都沒有刻度，所以你只能進行下列三種動作：

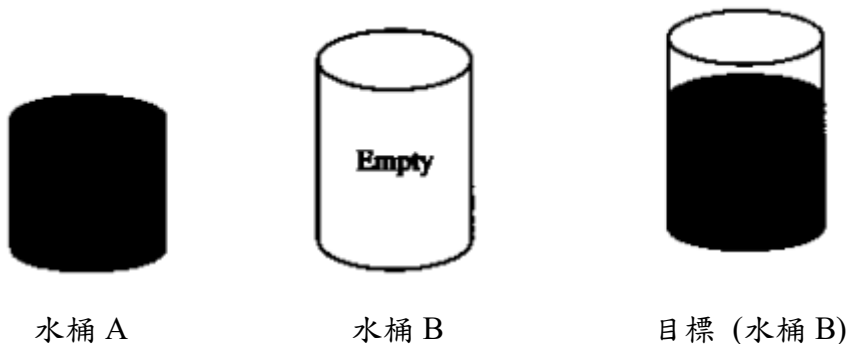
- (1) Fill 將水桶的水裝滿
- (2) Empty 將水桶的水倒光
- (3) Pour 將其中一個水桶的水倒到另一個水桶

其中，第三種動作僅有兩種可能，即第一個水桶的水須全部倒光、或是第二個水桶已裝滿便算結束。舉例說明，假設水桶 A 及水桶 B 都可容納 8 公升，若此時水桶 A 有 5 公升，水桶 B 有 6 公升，第一種動作可將水桶 A 裝滿，第二種動作可將水桶 A 倒光，第三種動作可將水桶 A 的水倒入水桶 B，但僅可將水桶 B 裝滿到 8 公升，使得水桶 A 剩下 3 公升。

水桶謎題的目的在使水桶 B 達到某給定的水量 (公升)，如圖所示為範例，若水桶 A 的容量為 3 公升，水桶 B 的容量為 5 公升，目標水量為 4 公升，則可達到目標的順序如下：

Fill A
Pour A B
Fill A
Pour A B
Empty B
Pour A B
Fill A
Pour A B
Success

其中，Pour A B 表示將水桶 A 倒水倒水桶 B 中。



注意：

1. 本題中你可以假設給定的謎題一定有解。
2. 水桶 A 與水桶 B 在剛開始時皆是空的。

輸入說明：

每組有三個數字，第一個數字為水桶 A 的容量，第二個數字為水桶 B 的容量，第三個數字為目標容量，單位均為公升。輸入為 0 0 0 時則結束。

輸出說明：

列出達到目標的順序。

輸入範例：

3 5 4

5 7 3

0 0 0

輸出範例：

Fill A

Pour A B

Fill A

Pour A B

Empty B

Pour A B

Fill A

Pour A B

Success

Fill A

Pour A B

Fill A

Pour A B

Empty B

Pour A B

Success

※ 本問題曾經出現在好萊塢電影「終極警探 3」。

