

# 中原大學資訊工程系 演算法分析第一次機測

Deadline: 4 / 7 / 2023 (星期五)  
(限期中考前測完，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，版本請採用目前最新版本(原則上，請直接下載與安裝 Anaconda)。
4. 可參考課本、參考書籍或網站資料等進行解題，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. 輸入與輸出採用標準格式或讀寫檔案方式進行。
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: 10927XXX / 10927XXX  
# 姓名: 陳○○ / 林○○  
# 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：

```
10927001_1.py  
10927001_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的輸入資料(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。程式的輸出結果錯誤、輸出格式與範例不符、或在執行後超過 5 秒(以每筆測資為基準)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

### 【執行時間測試】

機測預計採用個人電腦 CPU Intel i7、8G RAM、作業系統以 Windows 10 為主。建議同學在繳交程式前先使用下列 Python 程式進行初步的執行時間測試：

```
import time
start_time = time.time()
.....
total_time = time.time() - start_time
print(total_time)
```



## 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具有代表性的問題，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)的 Divide-and-Conquer 演算法，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 ( $1 \leq n \leq 10,000$ )，代表輸入的整數個數，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 ~ 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 (假設索引值從 1 開始)。

### 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
```

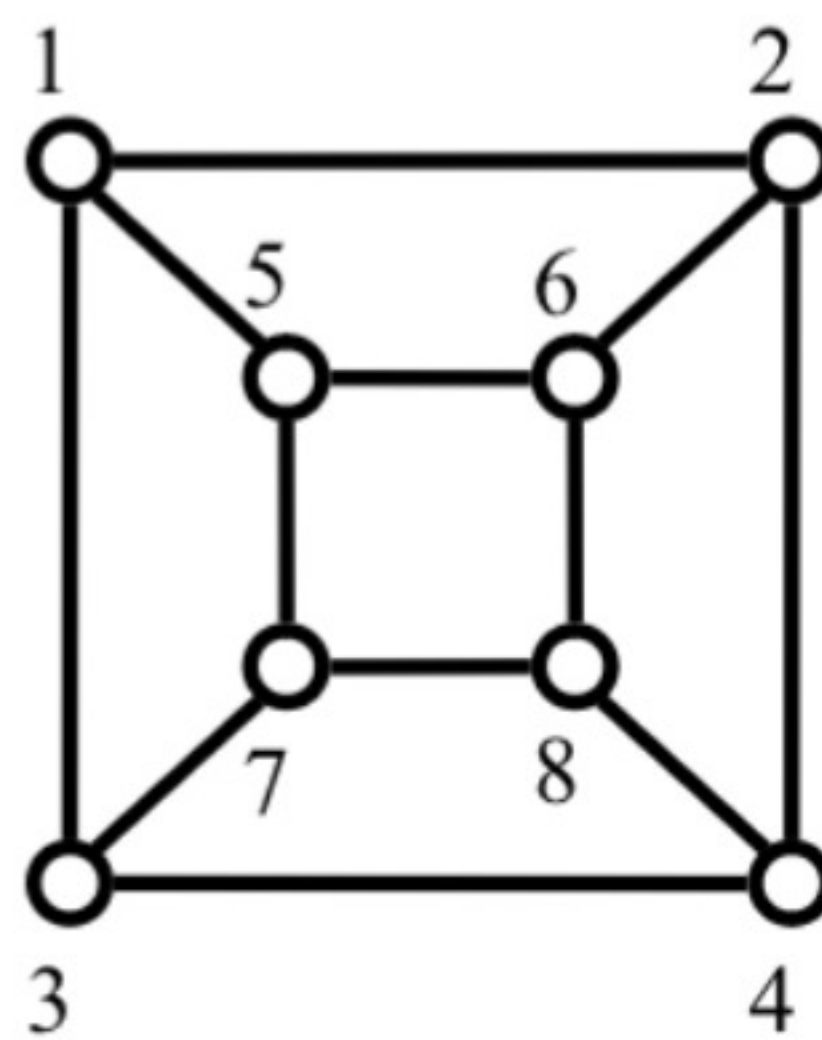
### 輸出範例：

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

## II. 漢密爾頓迴圈 (Hamiltonian Cycle)

給定一圖形  $G=(V,E)$ ，其中  $V$  代表頂點 (Vertex) 集合，依阿拉伯數字  $1, 2, \dots$  等安排； $E$  代表邊 (Edge) 的集合，依其連接的頂點定義。以下圖為例，則頂點的個數為  $|V|=8$ ，邊的個數為  $|E|=12$ 。

**漢密爾頓迴圈** (Hamiltonian Cycle) 的定義如下：從某一頂點出發（如頂點 1），陸續對其他頂點走訪，但每一頂點僅能走訪一次，且回到原出發點形成迴圈。本題中，將給定圖形，且漢密爾頓迴圈一定存在，試寫一程式列出其中一個漢密爾頓迴圈，且其出發點為頂點 1。



以上圖為例，則其中一個漢密頓迴圈為：1 2 4 3 7 8 6 5 1 (起點及終點均為頂點 1)。注意：本題中，答案不一定是唯一，但僅須列出其中一個漢密頓迴圈即可。

### 輸入說明：

第一行為頂點及邊的數目（以空格隔開），緊接依序為連接邊的兩個頂點，0 0 代表結束。注意：測試時可能包含多個圖形，但頂點的個數不會超過 10。

### 輸出說明：

列出其中一個漢密頓迴圈（分別以空格隔開），但起點及終點均為頂點 1。

### 輸入範例：(同上圖形)

```
8 12
1 2
1 3
1 5
2 4
2 6
3 4
3 7
```

4 8  
5 6  
5 7  
6 8  
7 8  
0 0

輸出範例

1 2 4 3 7 8 6 5 1



### III. 渡河問題 (River Crossing Problem)

假設有一主人有  $M$  隻狼與  $N$  隻羊 ( $M \leq N$ ) 需從西岸 (W) 渡河到東岸 (E)，其使用的船不含主人本身需乘載至少一隻才渡河，最多只能乘載兩隻。由於狼會吃羊的緣故，為了安全的考量，每次渡河時，不管在船上、西岸、或東岸，羊的數目均不能比狼少，除非羊的數目等於零。

若定義  $(X, Y, P)$  來表示渡河時的不同狀態，其中  $X, Y$  分別代表狼、羊在西岸的數目， $P$  代表渡河前船的位置 (W or E)。舉例而言，最初的狀態是  $(M, N, W)$ ，最後的狀態是  $(0, 0, E)$ ，其中一安全的狀態為  $(m, n, P)$ ，則  $m \leq n$  (除非  $n = 0$ )，且  $M - m \leq N - n$  (除非  $N - n = 0$ )，亦即不管在船上、西岸或東岸，羊的數目均不能比狼少。

試寫一程式解決渡河問題。現以下為範例，假設剛開始時，狼與羊各有兩隻 ( $M = 2, N = 2$ )，則其渡河的次序為 (在渡河次數最少的情況下)：

$$(2, 2, W) \rightarrow (0, 2, E) \rightarrow (1, 2, W) \rightarrow (1, 0, E) \rightarrow (2, 0, W) \rightarrow (0, 0, E)$$

其中渡河的次序中不應有重複。

#### 輸入說明：

每列代表輸入初始狼與羊的數目  $M$ 、 $N$  (均為正整數，且  $M \leq N$ )，0 0 表示結束。

#### 輸出說明：

顯示渡河的次序。

#### 輸入範例：

2 2

0 0

#### 輸出範例：

(2, 2, W)

(0, 2, E)

(1, 2, W)

(1, 0, E)

(2, 0, W)

(0, 0, E)

## IV. 小群體 (Small Groups)

一群人在一起時經常會形成一個一個的小群體。假設有  $n$  個人，編號由 0 到  $n-1$ ，每個人都寫下他最好朋友的編號 (好朋友有可能是他自己的編號，代表他自己沒有其他好友)。本題中，每個人的好友編號絕對不會重複，也就是說 0 到  $n-1$  每個數字都恰好出現一次。

這種好友的關係會形成一些小群體。例如： $n=10$ ，好友編號如下：

	0	1	2	3	4	5	6	7	8	9
好友編號	4	7	2	9	6	0	8	1	5	3

0 的好友是 4，4 的好友是 6，6 的好友是 8，8 的好友是 5，5 的好友是 0。所以 0、4、6、8、5 就形成了一個小群體。另外，1 的好友是 7，而且 7 的好友是 1。所以 1 和 7 形成另一個小群體。同理，3 和 9 是一個小群體。2 的好友是自己，因此他自己是一個小群體。因此，在這個例子中共有 4 個小群體： $\{0, 4, 5, 6, 8\}$ 、 $\{1, 7\}$ 、 $\{3, 9\}$ 、 $\{2\}$ 。

本題的問題是：輸入每個人的好友編號，計算出總共有幾個小群體。

### 輸入說明：

首先是一個正整數  $n$  ( $1 \leq n \leq 10,000$ )，代表團體中的總人數，0 則代表結束。接著，依序是 0 的好友編號、1 的好友編號、...、 $n-1$  的好友編號，共有  $n$  個數字，包含 0 到  $n-1$  的每個數字都恰好出現一次，數字間會有一個空白隔開。

### 輸出說明：

顯示小群體的個數。不要有任何多餘的字或空白，並以換行字元結尾。

### 輸入範例：

```
10
4 7 2 9 6 0 8 1 5 3
3
0 2 1
0
```

### 輸出範例：

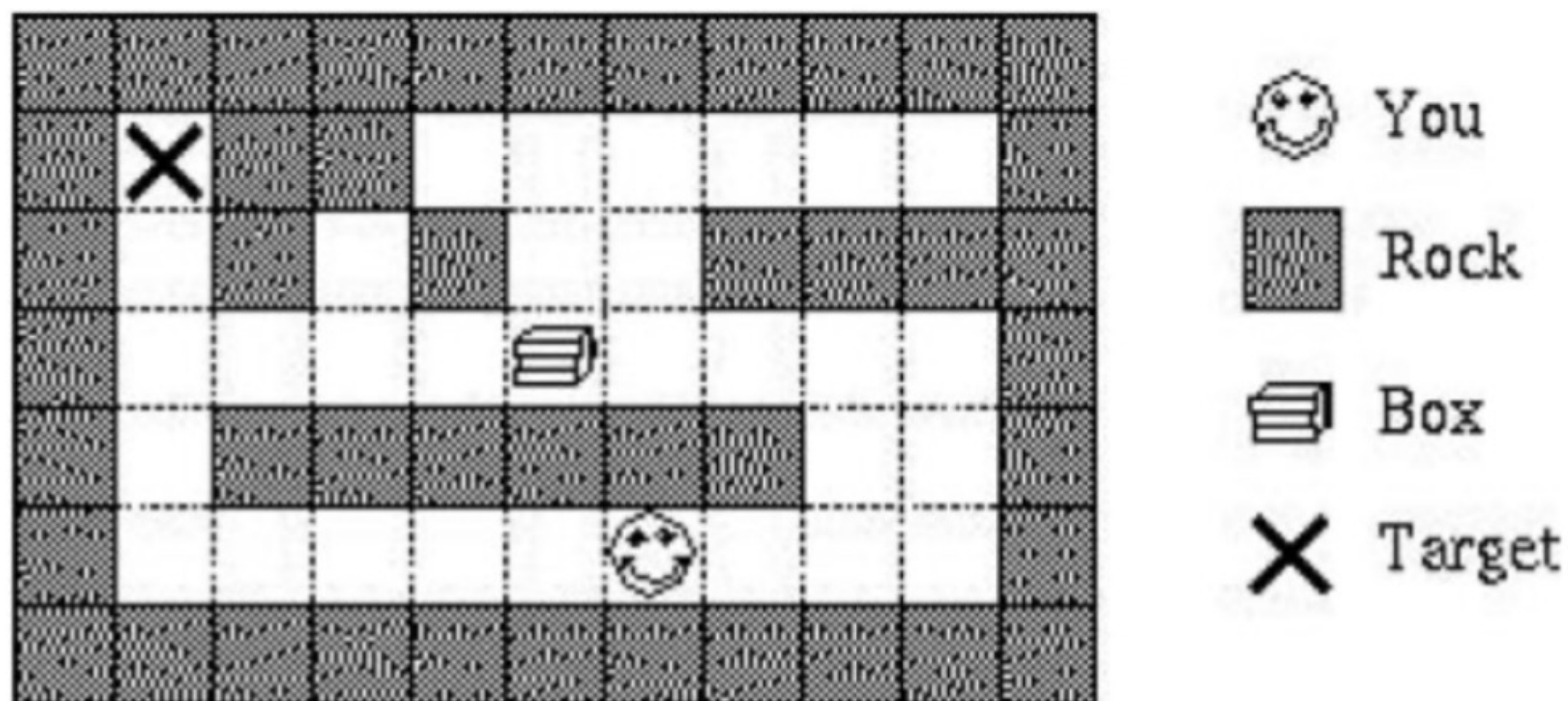
```
4
2
```



## V. 推箱子遊戲 (Pushing Box Game)

假設你身處在二維的迷宮中 (如圖)，迷宮可能含有 (或不含) **大石塊** (Rock)，在沒有大石塊阻擋時，你可以向東、西、南、北等四方向一次移動一格。其中一格擺了一個**箱子** (Box)，你只可以向箱子移動的方向推，譬如：若你站在箱子的東面，則你只能向西面推箱子；若你站在箱子的南面，則你只能向北面推箱子；以此類推。在任何情況下，由於箱子非常沉重，你都無法拉動箱子，萬一你把箱子推到死角，則你將再沒機會移動箱子。

如圖所示，有一個**目標** (Target)，你的工作是將箱子推到指定的目標，由於箱子非常沉重，因此推動箱子次數必須最少，試寫一程式解決這個問題。



### 輸入說明

輸入含有幾個迷宮，每一個迷宮首先定義迷宮的大小為  $r$  及  $c$ ，分別代表迷宮的列數及行數 (其中， $r, c \leq 20$ )。緊接為  $r$  列，每一列含  $c$  個字元 (Characters)，其中，大石塊用 # 表示，空格用 . 表示。你的初始位置為 S，箱子的初始位置為 B，目標的位置為 T。當輸入之  $r$  及  $c$  為 0 時代表結束。

### 輸出說明

首先，列出迷宮的編號。接著，印出一組推動箱子的方向。若無法將箱子推到指定的目標，則列出 Impossible。若有兩組推動箱子的次數均為最少，則列出總移動數最少的 (即含移動及推動的次數)。使用 E, W, S, N, e, w, s, n 分別代表推動或移動的方向 (東、西、南、北)，大寫表示目前需推動箱子，小寫表示僅需移動。迷宮與迷宮間則以空行隔開。

### 輸入範例

```
1 7
SB...T
1 7
```



SB..#.T  
7 11  
#####  
#T##....#  
#.#.#..####  
#....B....#  
#.#####..#  
#.....S...#  
#####  
0 0

本範例與上圖相同

輸出範例

Maze #1  
EEEEEE  
Maze #2  
Impossible  
Maze #3  
eennwwWWWWeeeeesswwwwwnNN