

Homework 2

TA mail: itai.mmcvlab@gmail.com

Problem - 工作指派問題(Job Assignment Problem)

- 我們定義工作指派問題為：假設有 N 件任務(task)要處理，手邊有 N 個人(agent)可以指派，每個人必須處理一件任務，每個任務也必須被一個人處理
- 我們定義成本(cost)為每個人在做每一個任務時所需要花費的時間，這個作業就是要請你設計演算法來找出如何分配人到不同的任務中使得的總成本最低

範例 Input

範例 input 為我們在這次作業中的 input 格式，input 為一個 $n \times n$ 的矩陣，每個 row 代表的是不同的 tasks，column 代表的是 agents

- `input[i][j]` 代表的是第 i 個 task 交給第 j 個 agent 時所需的時間
- `input` 中的數值資料型態為小數(float)

以下我們以這個例子來說明

```
input =  
[[10, 20, 23, 4],  
 [15, 13, 6, 25],  
 [2, 22, 35, 34],  
 [12, 3, 14, 17]]
```

以這個 input 來說，我們可以將它看成是以下表格，而紅色的值是將這個工作分配問題的最佳解，如果將 task0 分配給 agent3、task1 分配給 agent2、task2 分配給 agent0、task3 分配給 agent1 的話可以得到最小的時間成本(cost)

	agent0	agent1	agent2	agent3
task0	10	20	23	4
task1	15	13	6	25
task2	2	22	35	34
task3	12	3	14	17

範例 Output

```
yourAssignment = [3, 2, 0, 1]
Cost = 15
```

- `yourAssignment[i]` 的值代表第 `i` 個 task 要分給的 agent 是誰，以上面這個 assignment 為例，代表的是 task0 給 agent3 做，task1 給 agent2 做，以此類推
- 因為每個 task 只能給一個 agent 做，每個 agent 也只能做一個 task，所以 `yourAssignment` 中的數值不能重複且落於 `0~inputNum-1` 之間
- 這個範例 output 是這次作業的 output 要求，要請同學 print 出你的分配結果以及這個分配結果會使用的時間成本是多少

Problem class

以下的 class 提供給同學來使用，創建時可以將 input 丟入，這裡教同學來使用這個 class

```
class Problem:
    def __init__(self, input):
        self.input = input
        self.numTasks = len(input)

    def cost(self, ans):
        totalTime = 0
        for task, agent in enumerate(ans):
            totalTime += self.input[task][agent]
        return totalTime
```

創建 class

```
solver = Problem(input)
input = solver.input
numTasks = solver.numTasks
```

這個 class 提供兩個 attributes

- `input`: 每個 agent 在每個 task 中所需要花費的時間
- `numTasks`: tasks 的數量

```
cost = solver.cost(yourAssignment)
```

可以利用 `cost` 這個 method 來算出使用你的演算法得出的 assignment 所需要花費的時間成本為多少

作業要求

Coding problem1. Brute Force (暴力法) - 25%

- 用 python 寫出暴力法來暴力求解這個問題
- 將這個程式命名為 BF.py

Coding problem2. Genetic Algorithm (基因演算法) - 35%

- 用 python 寫出基因演算法來求解這個問題
- 基因演算法有很多不同的變形，例如
 - 如何將問題表示成染色體(chromosomes)
 - 每次交配(crossover)要設定幾個斷點
 - 突變(mutation)的機率
 - fitness函數的設計
 - 等等等等
- 以上不同的變形可以請同學自行嘗試，並將各種不同的實驗結果寫在 report 中的 **Section3**
- 將這個程式命名為 GA.py

Report - 40%

Section1. 如何實現 P1

Section2. 如何實現 P2

Section3. P2 實作過程中觀察到甚麼？

Section4. 比較兩個方法的優缺點，實作時的心得

-
- 以下程式提供給同學讓同學知道如何使用 problem 這個 class，同學可以依照這個格式並加入自己的演算法
 - 演算法可以寫在下方兩個區域，依照同學要將演算法設計成一個 class、function 或是直接寫在 main function 裡都可以

```

class Problem:
    def __init__(self, input):
        self.input = input
        self.numTasks = len(input)

    def cost(self, ans):
        totalTime = 0
        for task, agent in enumerate(ans):
            totalTime += self.input[task][agent]
        return totalTime

#####
# your algorithm class
#####

if __name__ == '__main__':
    input = [
        [10, 20, 23, 4],
        [15, 13, 6, 25],
        [2, 22, 35, 34],
        [12, 3, 14, 17]
    ]
    #####
    # your algorithm
    #####
    yourAssignment = [3, 2, 0, 1] # 用演算法得出的答案

    solver = Problem(input)
    print('Assignment:', yourAssignment) # print 出分配結果
    print('Cost:', solver.cost(yourAssignment)) # print 出 cost 是多少

```

範例測資

以下提供這三個範例測資來給同學測試自己的程式，有 input 並有正確答案來測試演算法的正確性，實際上提交作業後我們在批改時測試資料會抽換成我們的測試資料

```

input =
[[10, 20, 23, 4],
 [15, 13, 6, 25],
 [2, 22, 35, 34],
 [12, 3, 14, 17]]
# Output
Assignment: [3, 2, 0, 1]
Cost: 15

```

```

input =
[[0.43045255, 0.78681387, 0.07514408, 0.72583933, 0.52916145, 0.87483212, 0.34701621],
 [0.68704291, 0.45392742, 0.46862110, 0.67669006, 0.23817468, 0.87520581, 0.67311418],
 [0.38505150, 0.05974168, 0.11388629, 0.28978058, 0.66089373, 0.92592403, 0.70718757],
 [0.24975701, 0.16937649, 0.42003672, 0.88231235, 0.74635725, 0.59854858, 0.88631100],

```

```
[0.64895582, 0.58909596, 0.99772334, 0.85522575, 0.33916707, 0.72873479, 0.26826203],
[0.47939038, 0.88484586, 0.05122520, 0.83527995, 0.37219939, 0.20375257, 0.50482283],
[0.58926554, 0.45176739, 0.25217475, 0.83548120, 0.41687026, 0.00293049, 0.23939052]]
# Output
Assignment: [0, 4, 3, 1, 6, 2, 5]
Cost: 1.45020202
```

```
input =
[[0.71773280, 0.28980792, 0.86571783, 0.44026587, 0.53155829],
 [0.30556295, 0.56751479, 0.75442822, 0.62446877, 0.30992529],
 [0.83717620, 0.52213939, 0.54137934, 0.15001555, 0.70178034],
 [0.53279199, 0.08006661, 0.70693305, 0.29315974, 0.69018493],
 [0.03981310, 0.25511235, 0.94795653, 0.41611858, 0.50587076]]
# Output
Assignment: [3, 4, 2, 1, 0]
Cost: 1.41145021
```

測試資料形式

測試時會提供同學 input.json 檔案，裡面包含測資，形式為 dictionary，key 為題號，value 為測試的 input，舉例如下

```
{
  "0":
  [[0.26300727684204517, 0.48513471953446996, 0.8491417036699047],
   [0.7518785807425733, 0.029752222747783996, 0.5887209536993653],
   [0.7761974553100254, 0.19546118308946114, 0.6158427400193519]],
  "1":
  [[0.029027546749207156, 0.9596392488940668, 0.054290564347959425],
   [0.8792382327744873, 0.4842657230586963, 0.5985550149648665],
   [0.2892086218859313, 0.3626276202039942, 0.6700932067358675]]
}
```

如何將檔案轉為 input 可以參考以下程式碼

```
import json

with open('input.json', 'r') as inputFile:
    data = json.load(inputFile)
    for key in data:
        input = data[key]
```

Notice

- 請使用 Python 完成作業

- **不可直接使用 Python 提供的演算法，也不能使用一些外部的 library 來完成**
 - 但可以使用 python 內建標準函式庫，詳細請見 <https://docs.python.org/zh-tw/3.7/tutorial/stdlib.html>
 - 因為測試時會使用 json 檔案，可以用 json 函式庫
- 檔名請取 **BF.py GA.py**
- **禁止抄襲**
- 繳交格式：請將兩份程式和報告壓縮成 **.zip** 檔並命名為 **hw2_學號.zip**

```
hw2_P12345678.zip  
> BF.py  
> GA.py  
> input.json (測試時會替換)  
> report_P12345678.pdf
```