

# Homework3: Constraint Satisfaction Problem

CE6020 Artificial Intelligence (2022 Fall)

[ncu-ai-2022fall-ta@googlegroups.com](mailto:ncu-ai-2022fall-ta@googlegroups.com)

# Description

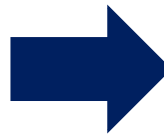
- 使用 Python3.x 解出以下約束滿足問題 (Constraint Satisfaction Problem)
  - **Q1:** Sudoku
  - **Q2:** Course Scheduling
  - **Q3:** N-Queens
- 請參考助教提供的範例程式，並在指定範圍內撰寫你的程式碼。

# Q1: Sudoku

- 每個數字 1-9 必須在每一行中恰好出現一次；每個數字 1-9 必須在每一列中恰好出現一次。
- 每個數字 1-9 必須在每一個 3x3 子框中恰好出現一次。
- 若有多組解，只需輸出其中一解即可。
- Constraints
  - $2 \leq \text{box\_size} \leq 3$
  - $\text{matrix}[i].\text{length} == \text{box\_size}^2$

```
{
  "box_size": 3,
  "matrix": [
    [5,3,0,0,7,0,0,0,0],
    [6,0,0,1,9,5,0,0,0],
    [0,9,8,0,0,0,0,6,0],
    [8,0,0,0,6,0,0,0,3],
    [4,0,0,8,0,3,0,0,1],
    [7,0,0,0,2,0,0,0,6],
    [0,6,0,0,0,0,2,8,0],
    [0,0,0,4,1,9,0,0,5],
    [0,0,0,0,8,0,0,7,9]
  ]
}
```

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



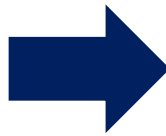
```
{
  "result": [
    [5,3,4,6,7,8,9,1,2],
    [6,7,2,1,9,5,3,4,8],
    [1,9,8,3,4,2,5,6,7],
    [8,5,9,7,6,1,4,2,3],
    [4,2,6,8,5,3,7,9,1],
    [7,1,3,9,2,4,8,5,6],
    [9,6,1,5,3,7,2,8,4],
    [2,8,7,4,1,9,6,3,5],
    [3,4,5,2,8,6,1,7,9]
  ]
}
```

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

## Q2: Course Scheduling

- 您需要修 `numCourses` 門課程，課程代號分別從 `0` 到 `numCourses-1`。
- `prerequisites` 列表會說明哪些課程有先修課程
  - E.g. `[1, 0]` 代表要若要修課程 `1`，您必須先修課程 `0`。
- 輸入 `numCourses` 與 `prerequisites`，回傳所有課程的修課順序。
  - 若有多組解，只需輸出其中一解即可
  - 若無解(不可能修完所有課程)，則回傳一個空陣列。

```
{  
  "numCourses": 2,  
  "prerequisites": [[1,0]]  
}
```



```
{  
  "result": [0,1]  
}
```

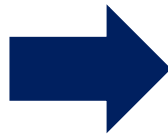
## Q2: Course Scheduling (cont.)

- Constraints
  - `1 <= numCourses <= 2000`
  - `0 <= prerequisites.length <= numCourses * (numCourses - 1)`
  - `prerequisites[i].length == 2`
  - `0 <= ai, bi < numCourses`
  - `ai != bi`

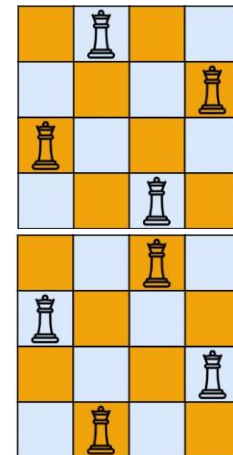
# Q3: N-Queens

- 西洋棋中的「皇后」可以在線性距離吃掉任何旗子。
- 給定一個正整數  $N$ ，而在此  $N \times N$  棋盤中擺放  $N$  個皇后棋，且彼此不能互相攻擊。
- 本題只能使用 **Backtracking Algorithm**，並需輸出所有可能解。
- Constraints
  - $1 \leq n \leq 9$

```
{  
  "n": 4  
}
```



```
{  
  "length": 2,  
  "result": [  
    [[0, 1, 0, 0],  
     [0, 0, 0, 1],  
     [1, 0, 0, 0],  
     [0, 0, 1, 0]],  
    [[0, 0, 1, 0],  
     [1, 0, 0, 0],  
     [0, 0, 0, 1],  
     [0, 1, 0, 0]]  
  ]  
}
```



# Submission

- 請將檔案請包成一個 zip 壓縮檔，並繳交至 ee-class 平台
  - 壓縮檔名格式為 學號.zip (e.g. 111525000.zip)
  - 在你的 學號.zip 內至少要有下列 3 類檔案
    - q1.py Q1主程式
    - q2.py Q2主程式
    - q3.py Q3主程式
    - Python 版本指定 3.7.x
    - 可使用原生套件，不得安裝額外套件
    - 各題執行時間皆須小於 1 分鐘
- **Deadline: 2022/11/24 23:59**
  - 遲交：本次作業成績 = 原始成績  $\times$  (1 - (遲交天數  $\times$  0.2))
  - 檔名錯誤：本次作業成績 = 原始成績 - 10分

# Grading Policy

- **Programming (100%)**

- $Score_{Qi} = A + B$

- $A = \begin{cases} 3, & \text{通過基礎測資} \\ 0, & \text{else} \end{cases}$

- $B = \begin{cases} 2, & \text{通過進階測資} \\ 0, & \text{else} \end{cases}$

- $ProgrammingScore(100\%) = (\sum_{i=1}^3 \frac{Score_{Qi}}{15}) \times 100$