CSIE 5400 - 人工智慧
Artificial Intelligence

HW0

Spring 2021

bit.ly/**AI21S-HW0**

Course Logistics

- NTU COOL: https://cool.ntu.edu.tw/courses/4810
- FB Group: AI@NTU (2021 Spring) [http://bit.ly/AI21S-FB]
- TA Email: aita2021s@agent.csie.ntu.edu.tw
- TA Hours:
  - Wednesdays: 16:00-17:00, CSIE Lab R344
- TAs
  - Chao-Chun Han (韓兆駿)
  - Jeng-Luen Yu (余政倫)
  - Yi-Ping Bai (白宜平)
  - Erick Chandra
- Q&A:
  - Post Questions on FB (preferred)
  - Email
  - Do not private message TAs, you may send email.

JOIN FB GROUP!

Setting Up NTU COOL

⚠️ IMPORTANT ⚠️

- For students who have already enrolled in this course, please enter NTU COOL website directly.
- For students:
  - Who hope to get the authorization code
  - Who hope be auditors
  - Who are non-NTU students and have not received course invitations

Please provide your email address in the following Google Sheets by **23:59 tomorrow (Feb 23)**. TA would send the invitation to you by Wednesday. Please follow the instructions in it to sign up NTU COOL and then you could hand in HW0. Google Sheets: https://reurl.cc/ZOaYnl

HW0 - Prerequisite to Enroll

This homework aims to provide you an exercise on coding and introduces you to be familiar with the next assignments.

本作業旨在為您提供編碼練習，並讓您熟悉下幾個作業。

Programming language **must** be in **Python 2.7**

程式語言**必須**使用 **Python 2.7**

Deadline 期限於 週日 **SUN, 28 FEB 2021 23:59** (UTC+8, server time 伺服器時間)

Caution: No late submission is allowed.

注意：遲交不得分

HW0 - Submission on NTU COOL (Assignments > HW0)

Deadline: FEB 28, 2021 23:59 (UTC+8, server time)

Caution: You will not be allowed to take this course for late submission or wrong answer

Programming language : **Python 2.7**

Filename: hw0_r012345678.zip
- hw0_r012345678/
  - hw0.py

Incompatible format will not be graded.

期限：2021年2月28日 23:59（UTC+8，伺服器時間）

注意：若您遲交或答錯，您無法繼續修習這門課

程式語言：**Python 2.7**

文件格式：hw0_r012345678.zip
- hw0_r012345678/
  - hw0.py

錯誤的格式將不會拿到分數。

HW0 - Determine if a Graph Is Bipartite

1. In this homework, you're required to write code in **Python 2.7** to determine if the input graph is a bipartite graph.
2. Input format:
   a. the first line of the input file contains one integer N indicating the number of vertices the graph has.
   b. In the following N lines, each line i contains several integers $v_1,...,v_m$ which means nodes $v_1,...,v_m$ are linked to the node i.
3. Output format:
   a. if the graph is bipartite: `True`
   b. else: `False`

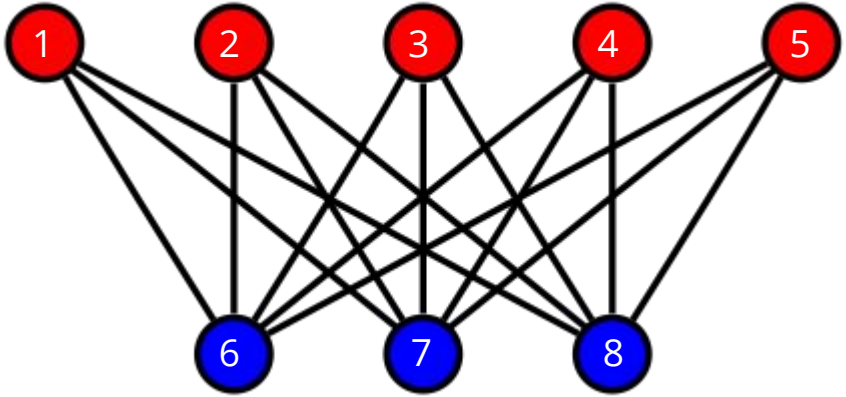
HW0 - Determine if a Graph Is Bipartite

- Sample Input:

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

- Output:

```
True
```



HW0 - Determine if a Graph Is Bipartite (Sample Code)

Please follow the class/function names exactly like depicted in the sample code.

請按照示範代碼中描述的函數名稱進行操作。

NB: A function must `return` **True or False**

Sample code link : https://reurl.cc/Q7XlWM

```
import sys

class Graph:

    def __init__(self, n):
        self.size = n
        self.adjacencyList = [[] for i in range(n+1)]

    def setEdge(self, u, v):
        self.adjacencyList[u].append(v)

    def bipartite(self):
        """Check if this graph is bipartite

        Returns:
          True/Fasle
        """
        #TODO
        pass


if __name__=="__main__":

    with open(sys.argv[1]) as f:

        lines = f.readlines()
        size = int(lines[0])
        graph = Graph(size)

        for i in range(1, graph.size+1):
            neighbors = [int(v) for v in lines[i].split()]
            for v in neighbors:
                graph.setEdge(i, v)

    print(graph.bipartite())
```

HW0 - Determine if a Graph Is Bipartite

1. TAs will run the following command to test your program:

```
$ python hw0.py input > output
```

2. Do not import any packages

Additional Warnings

- It’s an easy question set. Do not overthink or over complicate it. You just need to use the basic of Python and some fundamental logic.
- Your Python environment must be Python 2.7
- It’s recommended to use virtual environment (please Google the usage if you do not know how to use)
- Please google for the solutions before you come to TA.
- If you cannot make the TA hours, please email the TAs to schedule an appointment instead of stopping by the lab directly.