UDACITY

Verified Certificate Evaluation

STUDENT **Stephen Mather**

COURSE Introduction to Computer Science

Hi Stephen,

Congratulations! This is one of the best submissions I have seen, and it easily meets our specifications. Accordingly, I'll give you the type of feedback I would give if this were a professional code review.

Thanks for including a docstring on `create_data_structure` and `games_chart`. It would be great if every function had a docstring. Also in `create_data_structure`, string literals should be moved to constants. I would prefer the network to be stored as a dictionary of dictionaries containing sets:

```
network[user] = {
    CONN_KEY : set(connections.split(', ')),
    GAMES_KEY : set(games.split(', '))
}
```

When execution reaches the end of a function, there is an implicit `return none`. In `get_connections` and `get_games_liked` it would be best to remove the explicit `return none`.

In `add_connection`, I would prefer a guard instead of the present conditional:

```
if user_A not in network or user_B not in network:
    return False
```

In path_to_friend, I would prefer the base case to be:

```
if user_A == user_B:
     return[user_A]
```

Your MYOP could be simplified by the use of the Counter class in the collections module:

https://docs.python.org/2/library/collections.html#collections.Counter

Your code is clean and extremely readable, and your `create_data_structure` is the shortest I've seen that doesn't use regular expressions! Congratulations on learning so much Python and coding best practices!

Code Functionality

Exceeds Specifications

- Code completes the desired task and uses a data structure which allows for efficient execution of the code even with large inputs.

Use of Control Statements

Exceeds Specifications

- Selection of control statement is never inappropriate.
- Looping control statements are appropriately used to avoid repetitive code.
- Branching control statements are used correctly and logic is expressed concisely and clearly.
- Appropriate Python syntax is often used with control statements, making them more intuitive.

Use of Procedures

Meets Specifications

- Repetitive blocks of code are contained within their own procedures.
- Recursion is used correctly in the problem that asks for it.
- Choice of arguments and return values is aligned with the purpose of the procedure.

Make-Your-Own-Procedure (MYOP)

Meets Specifications

- MYOP is well planned, designed, and produces a useful result
- Comments clearly explain MYOP usage and functionality

Code Readability

Meets Specifications

- Most variables and helper procedures are given meaningful and useful names.
- Appropriate variables are used instead of hard-coded values, clarifying code logic.
- Comments are concise and to the point while still effectively explaining long procedures.
- Code is ready for personal review and neatly formatted.

PROJECT EVALUATION

Project Meets Specifications