(/)

Curriculum
Short Specializations ⌃
Average: **97.3%**   ⌄

0x0A. Prime Game

Algorithm   Python

👤 By: Carrie Ybay, Software Engineer at Holberton School

⚙ Weight: 1

📅 Project over - took place from Feb 12, 2024 6:00 AM to Feb 16, 2024 6:00 AM

☑ An auto review will be launched at the deadline

In a nutshell…

- **Auto QA review:** 14.0/16 mandatory
- **Altogether: 87.5%**
  - Mandatory: 87.5%
  - Optional: no optional tasks

For this project, you will need to leverage your understanding of prime numbers, game theory, and algorithm optimization to solve a competitive game scenario. The challenge involves determining the winner of a game based on the strategic removal of prime numbers and their multiples from a set of consecutive integers.

Concepts Needed:

1. **Prime Numbers**:
   - Understanding what prime numbers are.
   - Efficient algorithms for identifying prime numbers within a range.
2. **Sieve of Eratosthenes**:
   - An efficient algorithm for finding all prime numbers up to any given limit, which can be particularly useful for this task.
3. **Game Theory**:
   - Basic principles of competitive games where players take turns and the concept of optim play.

○ Understanding win conditions and strategies that lead to a win or loss.

(/) 4. **Dynamic Programming/Memoization**:

  ○ Using previous results to make future calculations faster, potentially necessary for optimizing the solution for multiple rounds of the game.

5. **Python Programming**:

  ○ Loops and conditional statements for implementing game logic and algorithms.
  ○ Arrays and lists for storing the integers and tracking removed numbers.

Resources:

- **Prime Numbers and Sieve of Eratosthenes**:

  ○ Khan Academy: Prime Numbers (/rltoken/IUKEfGVroNza8u37x0lEzw): Introduction to prime numbers.
  ○ Sieve of Eratosthenes in Python (/rltoken/sVjdrNQEaErO_qRYsVMTEg): A step-by-step guide to implementing the sieve algorithm in Python.

- **Game Theory Basics**:

  ○ Game Theory Introduction (/rltoken/lH4z--LnsuXYKh23Ji9Elw): A simple explanation of game theory and strategic decision-making.

- **Dynamic Programming**:

  ○ What Is Dynamic Programming With Python Examples (/rltoken/W6T0RxWaFG3GisPxLLNYkQ): An introduction to dynamic programming with Python examples.

- **Python Official Documentation**:

  ○ Python Lists (/rltoken/JTEGXnSDYDp8yblD9y86eg): Managing lists in Python, useful for tracking the game state.

By grasping these concepts and making use of the recommended resources, you will be well-equipped to approach the problem with a solid understanding of both the mathematical and programming challenges involved. The key to success in this project lies in applying efficient algorithms to manage the game's state and making optimal decisions based on the game's rules.

Additional Resources

- Mock Technical Interview (/rltoken/97_pUpfqKpBXGrqmXmT7Vw)

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `PEP 8` style (version 1.7.x)
- All your files must be executable

(/)

Tasks

0. Prime Game

Score: 87.5% (*Checks completed: 87.5%*)

Maria and Ben are playing a game. Given a set of consecutive integers starting from `1` up to and including `n`, they take turns choosing a prime number from the set and removing that number and its multiples from the set. The player that cannot make a move loses the game.

They play `x` rounds of the game, where `n` may be different for each round. Assuming Maria always goes first and both players play optimally, determine who the winner of each game is.

- Prototype: `def isWinner(x, nums)`
- where `x` is the number of rounds and `nums` is an array of `n`
- Return: name of the player that won the most rounds
- If the winner cannot be determined, return `None`
- You can assume `n` and `x` will not be larger than 10000
- You cannot import any packages in this task

Example:

- `x = 3`, `nums = [4, 5, 1]`

First round: `4`

- Maria picks 2 and removes 2, 4, leaving 1, 3
- Ben picks 3 and removes 3, leaving 1
- Ben wins because there are no prime numbers left for Maria to choose

Second round: `5`

- Maria picks 2 and removes 2, 4, leaving 1, 3, 5
- Ben picks 3 and removes 3, leaving 1, 5
- Maria picks 5 and removes 5, leaving 1
- Maria wins because there are no prime numbers left for Ben to choose

Third round: `1`

- Ben wins because there are no prime numbers for Maria to choose

Result: Ben has the most wins

```
carrie@ubuntu:~/0x0A-primegame$ cat main_0.py
(/)
#!/usr/bin/python3

isWinner = __import__('0-prime_game').isWinner


print("Winner: {}".format(isWinner(5, [2, 5, 1, 4, 3])))

carrie@ubuntu:~/0x0A-primegame$
```

```
carrie@ubuntu:~/0x0A-primegame$ ./main_0.py
Winner: Ben
carrie@ubuntu:~/0x0A-primegame$
```

Repo:

- GitHub repository: `alx-interview`
- Directory: `0x0A-primegame`
- File: `0-prime_game.py`

[ ] Done?   Help   Check your code   Ask for a new correction   QA Review