Home(/)

0x09. Island Perimeter

Algorithm Python
My Playthrough(/planning/me)

Weight: 1

Projects(/projects/current)

Project over - took place from Aug 26, 2024 6:00 AM to Aug 30, 2024 6:00 AM

An auto QA review will be launched at the deadline

QA Reviews I can make(/corrections/to_review)

Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)

In a nutshell…

- **Auto QA review:** 9.75/15 mandatory
- **Altogether: 65.0%**
  - Mandatory: 65.0%
  - Optional: no optional tasks

Curriculums(/dashboards/my_curriculums)
Concepts(/concepts)
Conference rooms(/dashboards/video_rooms)
Servers(/servers)

For the "0. Island Perimeter" project, you will need to apply your knowledge of algorithms, data structures (specifically matrices or 2D lists), and iterative or conditional logic to solve a geometric problem within a grid context. The goal is to calculate the perimeter of a single island in a grid, where the grid is represented by a 2D array of integers. Understanding how to navigate and analyze 2D arrays and apply logical operations to determine the conditions for perimeter calculation is crucial for this task.

Sandboxes(/user_containers/current)

Concepts Needed:

Tools(/dashboards/my_tools)

1. **2D Arrays (Matrices)**:
   - Accessing and iterating over elements in a 2D array.
   - Understanding how to navigate through adjacent cells (horizontally and vertically).
2. **Conditional Logic**:
   - Applying conditions to determine whether a cell contributes to the perimeter of the island.
3. **Counting Techniques**:
   - Developing a method to count the edges that contribute to the island's perimeter.
4. **Problem-Solving Strategies**:
   - Breaking down the problem into smaller tasks, such as identifying land cells and calculating their contribution to the perimeter.
5. **Python Programming**:
   - Nested loops for iterating over grid cells.
   - Conditional statements to check the status of adjacent cells.

Video on demand(/dashboards/videos)
Peers(/users/peers)
Discord(https://discord.com/app)
My Profile(/users/my_profile)

Resources:

- **Python Official Documentation**:
  - Nested Lists (/rltoken/8SPalOgoGDWQChVbct0p1g): Understanding how to work with lists within lists in Python.
- **GeeksforGeeks Articles**:
  - Python Multi-dimensional Arrays (/rltoken/IYcYmeVlCfF-F7Szn1fzfQ): A guide to working with 2D arrays in Python effectively.
- **TutorialsPoint**:
  - Python Lists (/rltoken/TZ8UtQaRxN5cFf8c1TB-rw): Explains how to create, access, and manipulate lists in Python, which is essential for working with a grid.
- **YouTube Tutorials**:
  - Python 2D arrays and lists (/rltoken/H7SwII_XYDpwYonNYKXQfg)

By understanding these concepts and utilizing the provided resources, you will be equipped to approach the problem methodically. You'll need to iterate over the grid, apply logical operations to identify the perimeter of the island, and account for the specific conditions described in the task. This project not only tests your algorithmic thinking but also reinforces your ability to manipulate data structures and apply logical reasoning to solve problems.

Additional Resources

- Mock Technical Interviews (/rltoken/9ZYjQgC9HvOLZiHxmgd89Q)

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)
- You are not allowed to import any module
- All modules and functions must be documented
- All your files must be executable

Tasks

0. Island Perimeter

mandatory

Score: 65.0% (*Checks completed: 100.0%*)

Create a function `def island_perimeter(grid):` that returns the perimeter of the island described in `grid`:

- `grid` is a list of list of integers:
  - 0 represents water
  - 1 represents land
  - Each cell is square, with a side length of 1
  - Cells are connected horizontally/vertically (not diagonally).
- `grid` is rectangular, with its width and height not exceeding 100
- The grid is completely surrounded by water
- There is only one island (or nothing).
- The island doesn't have "lakes" (water inside that isn't connected to the water surrounding the island).

```
guillaume@ubuntu:~/0x09$ cat 0-main.py
#!/usr/bin/python3
"""
0-main
"""
island_perimeter = __import__('0-island_perimeter').island_perimeter

if __name__ == "__main__":
    grid = [
        [0, 0, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0],
        [0, 1, 0, 0, 0, 0],
        [0, 1, 1, 1, 0, 0],
        [0, 0, 0, 0, 0, 0]
    ]
    print(island_perimeter(grid))
guillaume@ubuntu:~/0x09$
guillaume@ubuntu:~/0x09$ ./0-main.py
12
guillaume@ubuntu:~/0x09$
```

Repo:

- GitHub repository: `alx-interview`
- Directory: `0x09-island_perimeter`
- File: `0-island_perimeter.py`

Check submission | View results