Water Tank Problem

Problem Statement

The Water Tank Problem involves computing the units of water stored between blocks of varying heights. Given an array of block heights, the task is to calculate the total units of water trapped between the blocks.

For example, given the input array [0,4,0,0,0,6,0,6,4,0], the expected output is 18 units.

Solution Overview

The solution to the Water Tank Problem involves implementing a web application that allows users to input an array of block heights. The application then calculates the units of water trapped between the blocks and visually represents the solution using SVG shapes or a table view.

The solution utilises a Flask web framework in Python for the backend and Vanilla JavaScript, HTML, and CSS for the frontend. The Flask application handles the HTTP requests and rendering of the HTML templates, while the JavaScript code handles the user input and updates the SVG or table view based on the calculated solution.

The core algorithm used to solve the Water Tank Problem is the Two-Pointer Technique. It involves traversing the input array from both ends simultaneously to determine the maximum height on the left and right sides for each block. By comparing the maximum heights, the algorithm calculates the water trapped between the blocks and accumulates the total units.

Solution Components

The solution consists of the following components:

1. Frontend: The frontend is implemented using HTML, CSS, and JavaScript. It provides a user interface for entering the block heights and displays the visual representation of the solution.

2. Backend: The backend is implemented using Flask, a Python web framework. It handles the HTTP requests and renders the HTML templates. It also includes the core algorithm for solving the Water Tank Problem.

3. SVG Generation: The solution utilizes the SVG (Scalable Vector Graphics) format to represent the blocks and water visually. The SVG shapes are generated dynamically based on the calculated solution.

4. Two-Pointer Algorithm: The core algorithm for solving the Water Tank Problem is implemented using the Two-Pointer Technique. It determines the maximum heights on the left and right sides for each block and calculates the units of water trapped between them.

Running the Application

To run the Water Tank Problem web application, follow these steps:

1. Install Python: Make sure you have Python installed on your machine.

2. Install Flask: Open a terminal or command prompt and install Flask using the following command: pip install flask.
3. Create Files: Create two files in the same directory: main.py and index.html.

4. Copy Code: Copy the provided code for main.py and index.html into their respective files.

5. Run the Application: In the terminal or command prompt, navigate to the directory containing the files and run the Flask application using the following command: python main.py.

6. Access the Application: Open a web browser and visit http://localhost:5000 to access the Water Tank Problem web application.
7. Enter Input: In the input field, enter the block heights as a comma-separated list. For example: 0,4,0,0,0,6,0,6,4,0.

8. Generate Solution: Click the "Generate" button to calculate the solution and visualize it using SVG shapes.

9. View Output: The application will display the generated SVG table or shapes, along with the total units of water trapped between the blocks.

Conclusion

The Water Tank Problem web application provides a user-friendly interface for solving and visualising the units of water trapped between blocks. By leveraging Flask, SVG, and the Two-Pointer Technique, the solution efficiently calculates the solution and presents it in a visually appealing manner. The application can be further extended with additional features