Approach and Challenges in Solving the Water Tank Problem

Introduction:

The Water Tank Problem entails determining the amount of water held between blocks of varying heights. This report describes my approach for solving the problem and the hurdles I encountered while developing a web application with Vanilla JavaScript, HTML, and CSS.

Approach:

1. Understanding the Problem:

- I began by fully comprehending the problem definition and requirements.

- To get insight on the solution's intended behavior, I analyzed the given input sequence and its predicted result.

2. Function Design:

- I created a method to determine the amount of water trapped between the blocks.

- Iterating over the input sequence, the function determined the left and right maximum heights for each block.

- I computed the water units trapped at each block by comparing the heights.

3. Implementing the Frontend Solution:

- To create a web application, I used Vanilla JavaScript, HTML, and CSS.

- I designed an interactive interface for entering block heights and visualizing water-filled portions.

- To represent the blocks and water sections, I built blocks using open-source frameworks or implemented a table view.

4. Testing and Refinement:

- I performed extensive testing on the application to ensure accurate calculations and proper visualization.

- I refined the application based on user experience for improvements.

Challenges:

1. Understanding the function:

- At first, understanding the function to calculate the water units trapped between blocks was difficult.

- It necessitated a comprehensive examination of the problem statement and the division of it into smaller subproblems.

2. Visualization Techniques:

- The visualization part of the online application was difficult to implement.

- Choosing between SVG shapes and table views necessitated thinking about the user experience and ease of implementation.

3. Handling Input Validation:

- Validating user input and dealing with potential issues such as incorrect characters or non-numeric values proved difficult.

- Thorough error handling was required to ensure a pleasant user experience and prevent inaccurate input.

Conclusion:

Finally, solving the Water Tank Problem and creating a web application with Vanilla JavaScript, HTML, and CSS was a fun experience. Regardless of the difficulties encountered in understanding the method, visualizing the answer, and handling input validation. The implementation provided an excellent learning experience. The web application correctly calculates and displays the units of water trapped between blocks in an interactive and visually pleasing manner.

To Compute water trapped: https://marayyy.github.io/IO-Factory-Water-Tank-Problem/