Acequia Project 3 Documentation

Noah Casey and Pratik Ojha

Course: ECE 231

Solution Description:

In the StudentSolution.cpp file, I wrote the solveProblems() function to control the flow of water between regions based on live data during the simulation. The logic uses the state of each region (whether it is in drought, near flooding, or below its water need) to determine whether a canal should be opened and what the flow rate should be. I prioritize regions in drought, and I dynamically adjust the thresholds that allow water to be shared when a region is at least 85% full. The function tries to fill North from East early, and then let North and South help East once they are stable. I also use higher flow rates when a region is flagged as being in drought. This solution doesn't solve every possible simulation — some random setups are too unbalanced or too short on time — but it succeeds in most runs where at least one region starts with usable water and there is enough time for transfer.

How the Code Works

The solveProblems() function is structured as a simulation loop that runs until either all regions are solved or the time limit is reached. In each simulated hour:

Read the State of Each Region:

The function checks each region's water level, need, and whether it's in drought or flooding.

Evaluate Each Canal:

For each canal (A, B, C, and D), it determines:

Should the canal be open?

What flow rate should it use?

This is based on whether the source region has enough water and whether the destination region still needs it.

Priority Rules:

Canal D (East → North) is used early to help fill North.

Once North and South are ≥ 85% full, they are allowed to send water to others.

Canals A, B, and C open only when the source is healthy and the destination needs water.

Drought regions are prioritized with a full flow rate (1.0).

Simulation Step:

After canals are adjusted, manager.nexthour() is called to update all water levels and flags.

Loop Control:

The loop continues until either all regions are out of drought and flood, or time runs out.

This loop-based design allows for adaptive control of water flow based on dynamic simulation feedback each hour.