Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Computer Technologies Engineering
Analysis and design of Advance Algorithms

Situated Problem 2: Waterworks

Group #607

Team #4
Santiago Quintana Moreno A01571222

Miguel Ángel Álvarez Hermida A01722925
Alfredo Luce Morales A01772499
Benjamin Christian Blaga A01830797

Ing. Luis Salomón Flores Ugalde

Monterrey, Nuevo León, 5th of December 2025

- **Section 1: Graph Parsing and Data Loading**
    - Building the parsing stage taught me how essential it is to create a clean internal representation before applying any algorithms. Extracting nodes, edges, coordinates, and pump information allowed me to treat the system as a structured graph rather than a raw text file. I recognize that many failures in real software projects begin with poor data modelling. In this case, there were no meaningful algorithmic alternatives to consider, since the task required accurate interpretation rather than optimization. What mattered was designing a reliable format that would support the complexity of later computations. This experience reminded me that strong foundations make advanced techniques possible.
- **Section 2: Pipe length Calculations**
    - Computing pipe lengths showed me how simple mathematical operations can become the backbone of a larger computational pipeline. Using geometric distance ensured consistent behaviour across all later stages. This step also reinforced the idea that not every part of a system benefits from algorithmic experimentation. More complex distance formulations were unnecessary because the Euclidean metric already aligned with the physical interpretation of the pipes. I find value in learning when not to overcomplicate a task. Precision and consistency were the priorities, so alternative approaches were considered irrelevant. This clarity helped the rest of the project operate on dependable cost values.
- **Section 3: Sectorization**
    - Sectorization taught me how important it is to choose algorithms that respect real network structure rather than just spatial clustering. We initially considered methods like K means and spectral clustering, but both ignored connectivity and introduced unnecessary computational cost. Bellman Ford also proved unsuitable because there were no negative weights, and it would have increased runtime. Using the shortest path approach from multiple pumps provided a solution that reflected how water actually moves through a network. I learned that the best algorithm is the one that captures the constraints of the world it models while remaining efficient and explainable. This balance made sectorization both accurate and robust.
- **Section 4: Sink Identification**
    - Identifying the farthest node in each sector helped me appreciate how often the simplest method is the most appropriate. We briefly considered using a max heap or even sorting nodes by distance, but both approaches introduced extra computation without improving the final result. A linear scan offered the best mix of clarity, speed, and maintainability. As someone preparing for a career in software engineering, I value solutions that are easy to verify and unlikely to fail silently. This step also reminded me that algorithmic pipelines rely on clean intermediate outputs. The chosen approach produced a dependable sink for each sector and ensured that later flow calculations had accurate reference points.

- **Section 5: Maximum Flow**
    - Working on maximum flow calculations allowed me to understand how resources move through constrained systems and how residual graphs evolve. We evaluated options like Push Relabel, and capacity scaling variants, yet all of them introduced complexity that was unnecessary for the size of the network. The method We used offered a level of transparency that made debugging and reasoning more manageable. I want to build systems that behave predictably, especially in domains like networking and embedded software. This experience taught me that the right algorithm is often the one that developers can implement confidently while still maintaining acceptable performance.
- **Section 6: TSP**
    - Designing the water quality sampling route helped me understand how practical heuristics can outperform exact methods in real engineering contexts. I reviewed alternatives such as BFS algorithms. It presented drawbacks ranging from high complexity to slow convergence and limited reproducibility. Using a nearest neighbor construction followed by a two opt refinement produced strong results with minimal overhead. I appreciate solutions that are straightforward, explainable, and adaptable. This section showed me that a good approximation can be far more valuable than a perfect answer that arrives too late or is too difficult to maintain.
- **Section 7: Expansion**
    - The expansion phase taught me how to design systems that adjust gracefully as new elements are added. I examined the idea of full re sectorization and tree optimization, but both were far too expensive for single node insertions. Random attachment also failed because it produced inefficient and unrealistic connections. Instead, linking each new node to the nearest existing one allowed the network to grow incrementally while keeping computational costs low. As a future computer scientist, I see this as a model for real systems that must scale without constant recalculation. This approach balanced practicality with cost.