A black background with grey leaves

AI-generated content may be incorrect.

ENHANCEMENT two: algorithms and data structures

cs 499 milestone three: enhancement two narrative

A grey logo on a black background

AI-generated content may be incorrect.

June 1, 2025

Hannah rose Morgenstein

Professor Saba Jamalian

**Milestone Three Narrative – Algorithms and Data Structure**

**Artifact Overview**  
The artifact I selected for this enhancement is the buffer creation logic from a custom GIS pipeline tool originally developed during my IT 338 coursework in Spring 2025. This tool automates spatial exclusion workflows by creating buffer zones around human-made features—such as roads, trails, and campsites—and subtracting those zones from designated wilderness boundaries to identify preservation areas. Initially, the buffer logic relied heavily on ArcPy, a proprietary GIS library integrated with ArcGIS Pro. While functionally correct, the original implementation was procedural, tightly coupled to ArcPy, and lacked modularity, performance optimization, and flexibility for broader use.

**Why I Selected This Artifact**  
I chose this artifact because it offers a rich opportunity to demonstrate and deepen my expertise in algorithms and data structures. The original buffer logic used iterative, procedural geometry operations that could be optimized using more efficient spatial data structures and vectorized algorithms. Additionally, replacing ArcPy with open-source libraries like GeoPandas and Shapely aligns with modern, accessible development practices. This artifact allowed me to showcase skills in computational geometry, algorithm optimization, and parallel processing, while also expanding the artifact’s scalability and usability.

**How the Artifact Was Improved**  
The core enhancement replaced ArcPy-based logic with open-source alternatives: GeoPandas, Shapely, and Fiona. This transition required rethinking the spatial buffer algorithm to leverage vectorized spatial operations, significantly improving computational efficiency. I implemented batch buffering and merged geometries using GeoPandas’ vectorized methods and Shapely’s unary\_union operation to reduce processing time. To further optimize performance, optional geometry simplification was introduced, balancing the trade-off between spatial accuracy and computational load. Multiprocessing was incorporated to parallelize buffering operations across large datasets, though this presented challenges in debugging and cross-platform compatibility. The codebase was modularized by separating buffer processing, data loading, and output writing into dedicated modules, enhancing readability, maintainability, and testing. Logging and error handling were added to improve robustness, and validation checks ensured spatial integrity of outputs.

**Outcome Alignment and Skill Demonstration**  
This enhancement directly supports the Computer Science program outcome:

*“Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices (data structures and algorithms).”*

The enhancement also aligns with the specific outcome I identified in Module One, focusing on designing efficient computing solutions with appropriate data structures and algorithms. I confirmed that this enhancement fully met the planned outcomes and did not require updates to my outcome-coverage plan. Through this project, I demonstrated practical application of algorithmic problem-solving by redesigning the buffer creation process with vectorized spatial computations, efficient spatial data structures, and optional multiprocessing to improve throughput. Using GeoDataFrames and geometry objects, I leveraged data structures optimized for spatial operations. I carefully managed trade-offs such as simplifying geometries to reduce complexity without significant loss of accuracy. These improvements illustrate my growing ability to design and evaluate scalable computing solutions, balancing performance, maintainability, and accuracy. Additionally, modular design and clear algorithm documentation improve the artifact’s professional quality and usability.

**Reflection on the Enhancement Process**  
This enhancement deepened my understanding of geospatial data structures and Python optimization techniques. I learned how GeoPandas and Shapely represent and manipulate geometry objects efficiently, which differs significantly from ArcPy’s approach. Implementing multiprocessing on Windows systems was challenging, requiring careful management of parallel processes and debugging strategies to maintain data integrity. Handling coordinate reference system (CRS) consistency and validating geometry correctness were crucial to ensure accurate spatial analysis.

Testing the enhanced artifact with real-world datasets confirmed performance gains and correctness; I used runtime benchmarks and visual inspection in GIS software for validation. Writing modular code and clear pseudocode helped me structure the implementation logically, facilitating debugging and future maintenance. Through this process, I learned how to critically evaluate trade-offs when balancing computational efficiency with accuracy and clarity. I also recognized the importance of ongoing refinement to meet evolving project needs, which has influenced how I approach algorithm design and data structure choices in other projects.

**Future Enhancements**

Looking ahead, I plan to incorporate security measures, such as input data validation and safeguards against data tampering, to enhance the tool’s reliability. Further scalability improvements could include distributed processing or integration with cloud services. Continued refinement of spatial algorithms and support for additional file formats will broaden the artifact’s applicability.