CS-499-18349-M01 Computer Science Capstone 2025

Adam Vosburg

Southern New Hampshire University

Milestone 3 (Enhancement Two) - Data Structures and Algorithms

03/23/2025

**Milestone 3 (Enhancement Two) - Data Structures and Algorithms**

The Grazios Rescue System is a comprehensive Node.js application designed to manage animal rescue operations. As part of my ongoing enhancements to this system, I focused on improving the algorithms and data structures used for animal matching, which represents a critical function in animal rescue operations - pairing the right animals with the right adopters or service needs. This enhancement was completed in March 2025 as part of my continuous improvement efforts for the system originally developed in early 2023.

I selected this enhancement for my ePortfolio because it showcases my ability to design and implement sophisticated algorithms and data structures to solve real-world problems. The animal matching system demonstrates my proficiency with JavaScript, algorithm design, and optimal data structure selection. The matching functionality represents a complex computational challenge: efficiently evaluating multiple animals against varied criteria with different priorities and weights. By implementing three distinct matching algorithms and a custom priority queue data structure, I've demonstrated my ability to analyze problems and select appropriate algorithmic approaches based on specific use cases and performance requirements.

The enhancement involved creating a new animal matching service that implements a weighted attribute matching algorithm, a service-specific matching algorithm, and a priority queue-based matching algorithm. The weighted attribute algorithm calculates match scores based on how well animals match specified criteria, with customizable weights for different attributes. For numeric attributes like temperament ratings, the algorithm uses a similarity calculation that provides proportional scores rather than binary matches. For array attributes like specializations, it calculates overlap percentages. The service-specific algorithm applies specialized scoring logic for different service animal types, evaluating attributes particularly relevant to service work. The priority queue implementation maintains animals in priority order based on match scores, efficiently retrieving the best matches.

This enhancement directly addresses the course outcome to 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. The implementation demonstrates algorithmic principles through custom matching algorithms, computer science practices through proper modular design, and trade-off management through algorithm selection for specific use cases. Additionally, the enhancement supports the "security mindset" outcome through careful input validation and the "professional-quality communications" outcome through comprehensive documentation of the algorithms and APIs.

The enhancement process provided significant learning opportunities and challenges. The weighted scoring algorithm required careful consideration of normalization techniques to ensure fair comparison across different attribute types. For example, comparing a boolean attribute like "ridingSafe" with a numeric attribute like "temperament" requires thoughtful scoring approaches to maintain fairness in the overall match score. The priority queue implementation challenged me to think about data structure selection trade-offs. While a heap-based implementation would offer theoretical advantages for large datasets, the ordered array approach I selected provides excellent performance for the expected data volume while maintaining code readability. One particularly challenging aspect was balancing algorithm complexity with usability. I wanted the matching system to be sophisticated enough to provide truly valuable matches while keeping the API simple enough for practical use. This required careful design of the input validation schemas and thoughtful defaults.

During the enhancement process, I gained a deeper appreciation for algorithm design and its impact on software performance and capability. I learned that effective algorithm design isn't just about theoretical efficiency but also about practical considerations like maintainability and extensibility. The priority queue implementation taught me how to evaluate when a custom data structure is warranted versus using built-in language features. By creating a tailored solution for animal matching that considers multiple attributes with varying importance, I've developed a more nuanced approach to algorithm design that I now apply to other complex problems. The animal matching system enhancement significantly improves the Grazios Rescue System's capabilities by providing sophisticated, efficient algorithms for pairing animals with adopters or service roles, demonstrating my ability to apply computer science principles to create practical, valuable solutions.