Jeffrey Karl

CS-499

29 September 2024

Milestone Three: Algorithms and Data Structures Narrative

The artifact I am presenting is a course advising program, which I developed during my Data Structures and Algorithms course in Winter 2024. Originally, the program was designed using a Binary Search Tree to store and manage courses. However, as part of my enhancement, I refactored the code to use a Java ArrayList in a sorted manner, demonstrating the flexibility of data structures and the importance of evaluating algorithmic efficiency based on specific needs. This change allowed me to manage the trade-offs involved in choosing between performance and simplicity, highlighting my ability to design computing solutions that are adaptable to various contexts.

I chose to include this artifact in my ePortfolio because it represents my problem-solving skills and algorithmic thinking when faced with design trade-offs. The program showcases my ability to balance complexity and performance by selecting the most appropriate data structure for a given problem. The transition from a BST to a sorted ArrayList enabled me to evaluate the strengths and weaknesses of each approach. While the BST offered better search efficiency for large datasets (O(log n)), the ArrayList allowed for simpler insertion logic and random access (O(1)) while maintaining acceptable performance (O(log n) for searches in a sorted list). This artifact demonstrates my proficiency in designing algorithms and managing trade-offs between ease of use and performance, a crucial aspect of real-world computing solutions.

This artifact highlights my understanding of both tree-based structures and list-based structures. Initially, the BST provided efficient logarithmic search capabilities, showcasing my ability to implement more complex data structures. However, the program's enhancement to use a sorted ArrayList demonstrates my skill in refactoring code to optimize for simplicity when performance constraints are not as critical. This transition illustrates my flexibility in adapting solutions to fit the specific needs of a project while maintaining overall functionality.

The program was improved significantly by transitioning to a sorted ArrayList, making it more accessible and easier to maintain for future modifications. The use of a sorted ArrayList allowed me to maintain efficient searches while simplifying insertion operations. This balance between performance and ease of use is particularly well-suited for small to medium-sized datasets, which aligns with the requirements of the project. The updated structure also reduced the complexity of the code, making it more understandable for other developers and ensuring it can be easily maintained and extended in the future.

In Module One, I set out to enhance my understanding of algorithm efficiency and the decision-making process involved in selecting appropriate data structures. This artifact enabled me to meet that goal by directly comparing the BST and ArrayList data structures, assessing their trade-offs, and implementing a solution that balances performance and maintainability. Through the enhancement process, I demonstrated my ability to design efficient, maintainable code while also improving my commenting and documentation practices. This focus on clarity and communication ensures that the code can be understood and modified by others.

Reflecting on the process of enhancing and modifying this artifact, I learned valuable lessons about the importance of code readability and maintainability. One challenge I faced was balancing the performance benefits of the BST with the simplicity and flexibility of the ArrayList. I learned that, in some cases, a simpler data structure like the sorted ArrayList can be more appropriate when working with smaller datasets or when ease of future modifications is a priority. Additionally, I refined my understanding of the importance of clear documentation, which is essential not only for myself but also for future developers who may need to work on the code. By focusing on efficient search logic through binary search in the ArrayList, I ensured that the program remained both functional and efficient.

This artifact allowed me to design a computing solution that meets the requirements of Course Outcome #3, demonstrating my ability to apply algorithmic principles and manage design trade-offs effectively. It showcases my skills in evaluating different data structures, optimizing for both performance and simplicity, and creating a maintainable solution that is flexible enough to meet the evolving needs of users.