

Homework 5: Reflection

1. **In retrospect, what could you have done better to reduce the time you spent solving this assignment?**

I could have approached the lexicographical ordering issue more methodically from the beginning. Rather than making multiple changes to the sorting logic, I should have first added detailed debug logging to verify exactly how the courses were being compared. I spent considerable time trying different implementations of the `compareTo` method and sort order, when the issue might have been in how the `hw4.Graph` was storing or returning the data.

I also should have isolated the specific test case earlier by creating a minimal example dataset and test. This would have allowed me to identify the root cause faster - a subtle issue with string comparison when handling underscores in course names. Unit testing individual components like the `PathSegment` comparator separately would have caught this sooner.

Finally, I should have been more diligent about examining the test reports in detail from the beginning. The error reports clearly showed the expected vs. actual output, which would have guided my debugging efforts more efficiently.

2. **What could the Principles of Software staff have done better to improve your learning experience in this assignment?**

The assignment could have included more detailed specifications about the expected lexicographical ordering behavior, particularly for edge cases like strings with special characters or underscores. A more comprehensive test suite with examples of the expected behavior for various edge cases would have been helpful.

Additionally, providing more guidance on how to debug issues like the one with lexicographical sorting would have been valuable. Perhaps a debugging guide or tutorial focused on common issues in graph implementations would help students approach problems more systematically.

The assignment could also have included more incremental milestones with automated feedback. For example, having separate checkpoints for the parser implementation, graph construction, and path-finding algorithm would have made it easier to identify where issues were occurring.

3. **What do you know now that you wish you had known before beginning the assignment?**

I wish I had a better understanding of how Java's string comparison works, especially with respect to special characters and underscores. The issue with `C_HIGH` being sorted before `C_LOW` despite the lexicographical order would have been easier to address if I had realized from the start that I needed to test string comparison behavior carefully.

I also wish I had known about more effective debugging techniques for graph algorithms. Adding strategic debug logging in the BFS implementation earlier would have saved considerable time. Understanding how to use JaCoCo for targeted coverage analysis would have also helped focus my testing efforts.

Finally, I wish I had realized the importance of designing for testability from the beginning. My implementation would have benefited from being more modular, with clear separation of concerns between parsing, graph construction, and path-finding. This would have made it easier to isolate and fix issues like the lexicographical sorting problem.