Extending Data Structures - Written Supplement

Q1

Question: If you updated your UML diagram(s) with different timing as last time (first instead of after, or after instead of first), what differences did you notice this time around? Justify your response.

Answer: In the original Data Structures assignment, I made my UML diagram before writing the program. However, I reversed the order this time as I believed the modifications would be minor enough to not warrant planning them out in a detailed fashion. A few differences that I noticed include more flexibility, more clarity while setting up relationships in the UML diagram, and difficulty in configuring coordination between classes. First, I had more flexibility. When I created the UML diagram first in the previous assignment, I noticed that a few of the relationship structures that I had originally planned either didn't make sense or were unnecessary when actually implemented, leading to changes within the diagram after implementation. However, in this assignment, since I wrote the code first, I had already iterated on it, making sure that the class structure was as efficient and clear as possible before putting it into the UML diagram. This made sure that I would only have to make the UML diagram once. Similarly, since I had written the code beforehand, I had more clarity on how each class interacted with each other, instead of wondering about how they should theoretically work with each other as I did in my first assignment. Finally, while it was quicker and faster to make the UML diagram, I noticed that it was a bit more complicated to imagine and remember how my new methods and classes interact with each other compared to putting it into a UML diagram first. As such, I would sometimes be stuck on the code trying to think of the best way to set the class structure up. While the benefits in terms of speed outweighed the drawbacks for this assignment, this was mainly because my changes were relatively minor compared to making the entire class structure from scratch. If my modifications required a significant reworking of the class structure or large-scale changes, I would likely have stuck to planning out my changes first in the UML diagram.