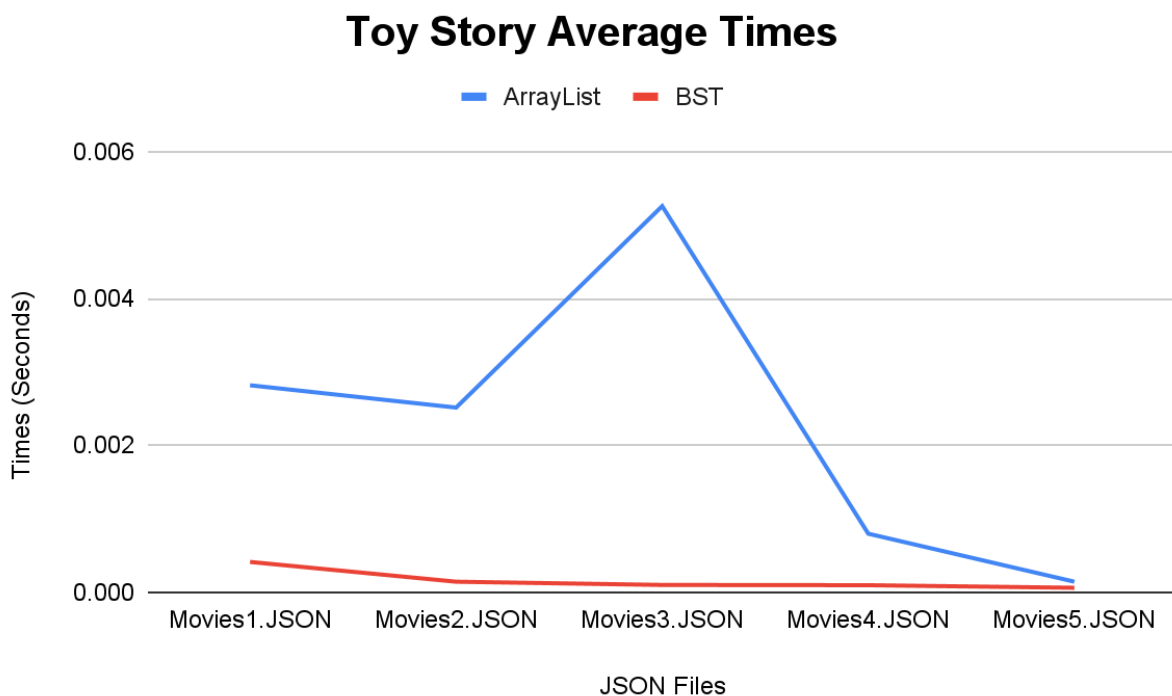
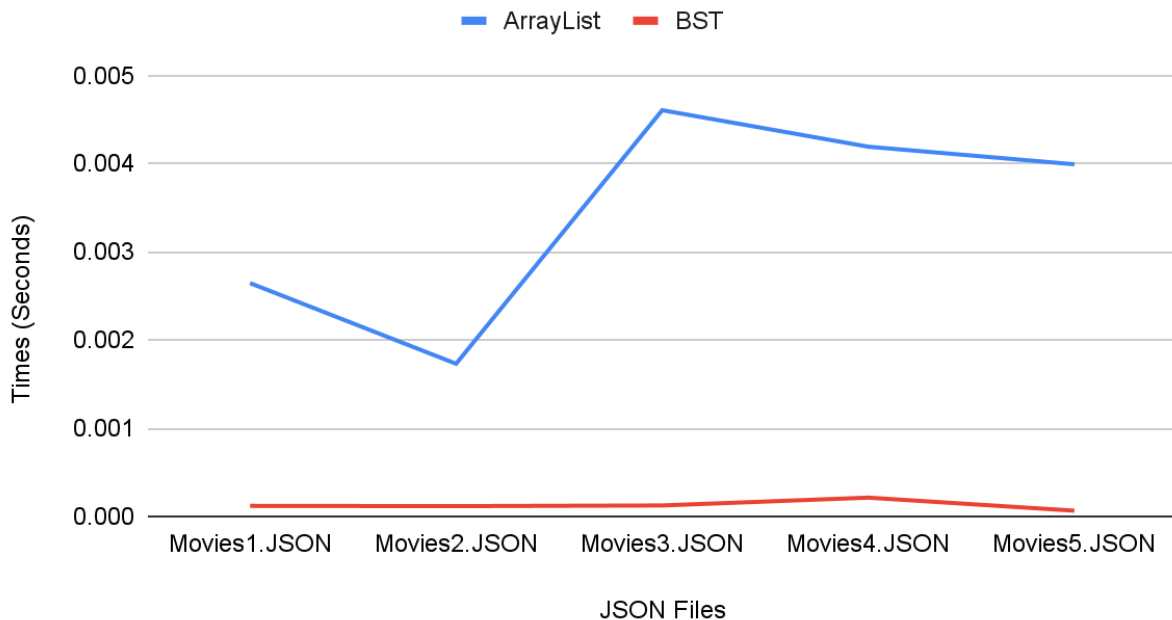


1. The height of Movies5 is 41 and the height of Movies3 is 32.
2. If MovieBST was balanced, searching for movies would become even faster than it already is, because each movement along the tree would eliminate half of the total options left.
3. I will use graphs to illustrate the time difference between the Binary Search Tree (BST) and the ArrayList. For the first graph, I searched for Toy Story in both structures three times and graphed the averages.



As you can see, the line for BST was not only straighter, showing a consistency in times produced, but was also lower than the ArrayList at every possible point. To ensure that this was not just due to searching “Toy Story”, I repeated the process with “Pikachu” instead.

Pikachu Average Times



As you can see, the trend remains the same, with BST consistently and greatly beating out ArrayList in both time and consistency.

4. The thing I am most proud of on this assignment is my ability to figure out how In Order works. At first it seemed to be a daunting task, but using logic and breaking down the problem made implementing InOrder not as hard as I thought it would've been.
5. Based on the data above, I would think it is fairly obvious that Binary Search Trees are the better choice for creating a movie database.
6. External Help:
None