

The assignments in this module helped me to focus on the space and time complexity of code, and how it is processed by the computer. In particular, I compared the time and space complexity of some of the standard built in Python methods and functions. I considered different sorting algorithms and the costs and benefits of using them. Although it is often easier to work with a sorted list, using a sorting function is not necessarily the most efficient way to program an algorithm.

In the first assignment I thought through how to implement my for loop as a recursive function, intending to reduce the time complexity of my code. However, in the end the more streamlined code which did not include recursion was faster. I realized I should also consider eliminating any unnecessary operations to quicken processing.

Additionally, I considered which type of objects I was using to store information in the programs. Sets are generally faster than lists, and using deque objects can be even faster as you can remove elements from either end. Looking at my partners' code I reflected on whether a hash or deque object could have been applied in the first assignment to increase efficiency. Overall, it seems that the most effective code is often the option that is the simplest and most elegant.