Daniel Gutierrez

Finding One

With the given instructions I decided to use a linear search approach even though it was strictly mentioned to avoid it. I chose it because it would assist my general understanding of what I was trying to build. The task was to write a program to find a unique element in an array where every other element appears exactly twice and provide meaningful comments on the code. I will say that before this assignment I had never put so much detail into comments and now I don’t think I’ll ever stop. Adding comments that explain in plain language what I just typed has greatly increased the amount of information I feel I am retaining while writing code.

My greatest strength of visualizing images is not doing so good when it comes to picturing all the iterations and loops. Understanding the logic of how the program should function was challenging for me. I used various resources such as YouTube, geeksforgeeks.org, Programiz.pro, and class notes to bolster my fundamental understanding of Java syntax and the algorithms.

Although it is a linear search and it’s not known for the best time and space complexity, I was able to add some efficiency in the code. If the conditions for the else block are met then the program will increment i, effectively skipping the next index. One other minor but efficient change (to some degree I’m sure) was applied at the very creation of the for loop. Iterating to the array length - 1 will end our search one index sooner. However, exactly how much efficiency or how those changes affected its time and space complexity, I don’t know. The best case time complexity would be if we found the single element at index one, O(1). The worst case is if the element is at the end of the array, O(N).