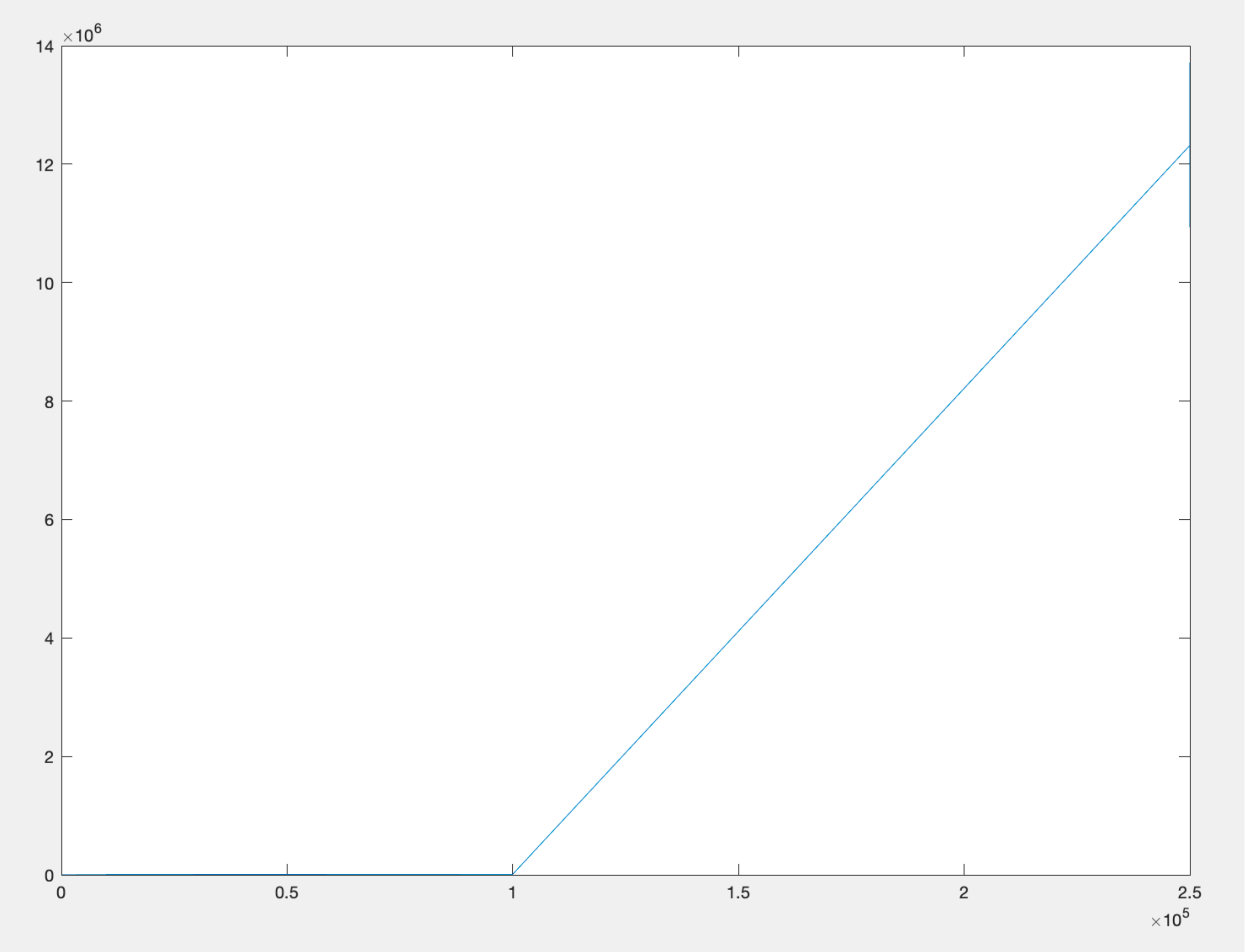
Below are the charts made using MATLAB for each of the three search algorithms.

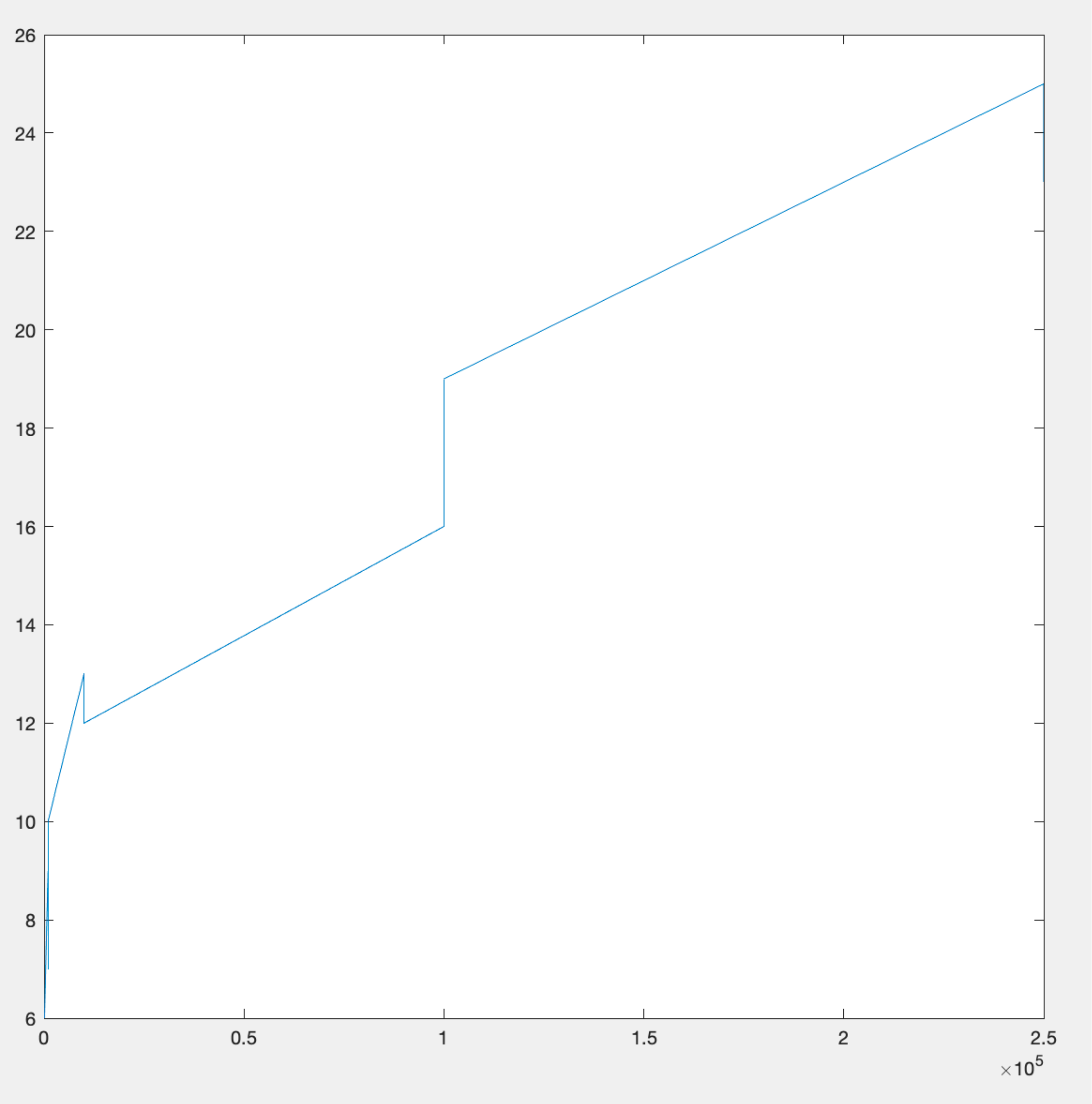
To complete this problem, I started by writing the three different algorithms as their own static methods in different classes. I then made another class with a static method which generates an ordered array of integers of the dictated size. Then, in my main method, I created arrays of the given sizes and pulled a value from each in order to give my algorithms something to be tested with. I tested each algorithm three with all of the arrays I made and wrote out messages to make it clear what was being tested and how many steps it took. Messages as for how many steps were implemented in the algorithms themselves.

Linear Search Algorithm Plot:



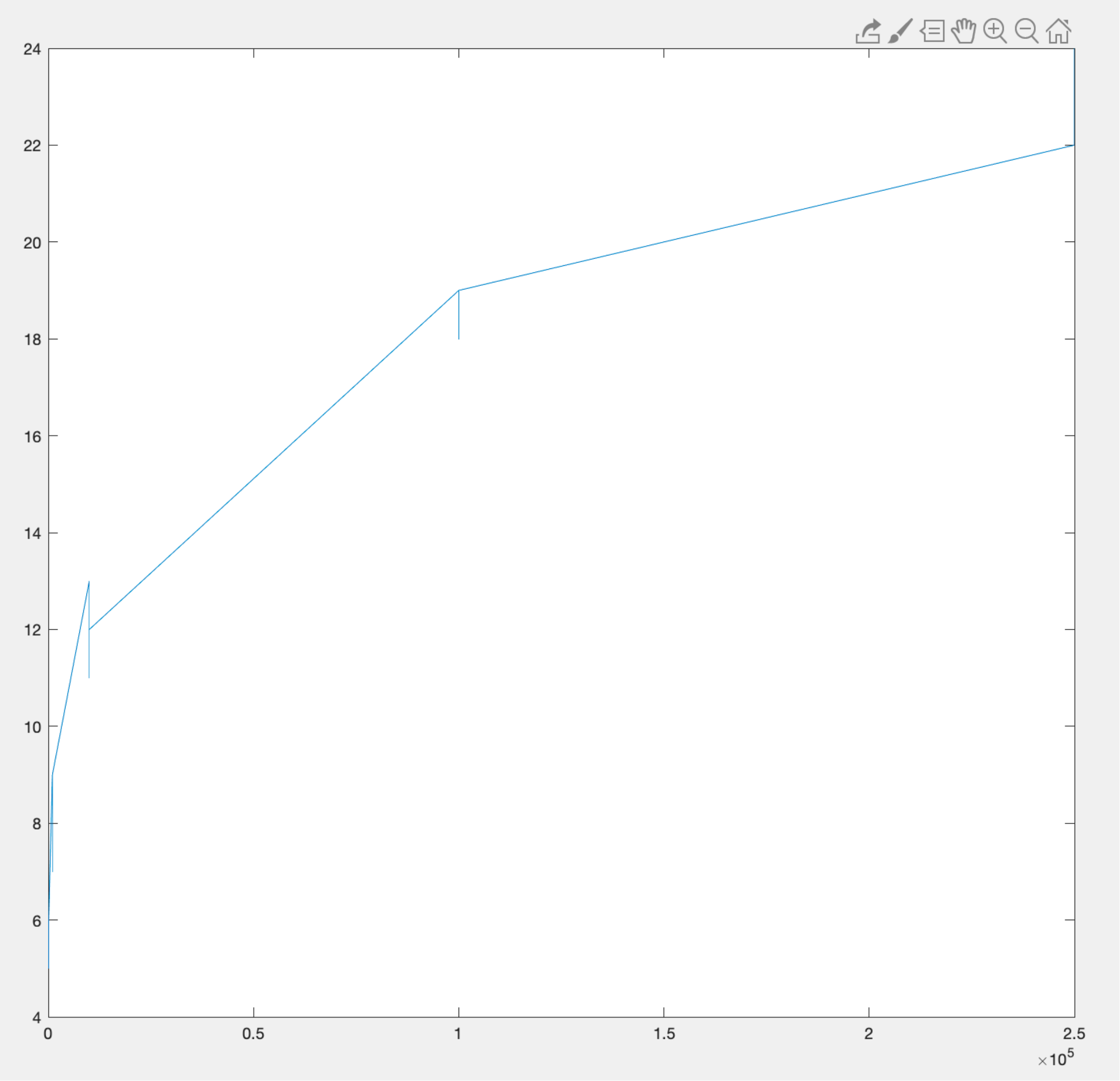
The linear algorithm is the most simple and least efficient. It merely searches each element one at a time. There are a few advantages to this however, one being that the array does not need to be sorted for the algorithm to work properly. The number of steps increases directly as the amount of data increases. It has a complexity of O(n)

Binary Search Algorithm Plot:



The next one is the Binary Search Algorithm. This one is much more efficient. It uses a while loop to keep breaking inspecting different half's of the function until it either hones in on the exact value or finds that the value is not there. As you can see form the graph, this one is much more efficient. As the data increases, the amount of steps does not increase that much. It has a complexity of O(log n)

Binary Recursive Search Algorithm Plot



The final algorithm is the Recursive Binary Search Algorithm. This algorithm works very much like the Binary Search except it makes use of recursion. Instead of a 'while' loop, the function utilizes recursion. The result is that the complexity and efficient is very simulate to the standard Binary Recursion. Because the function utilizes the powerful tool of recursion, the function body is able to be a more concise. It too has a complexity of 0(log n).