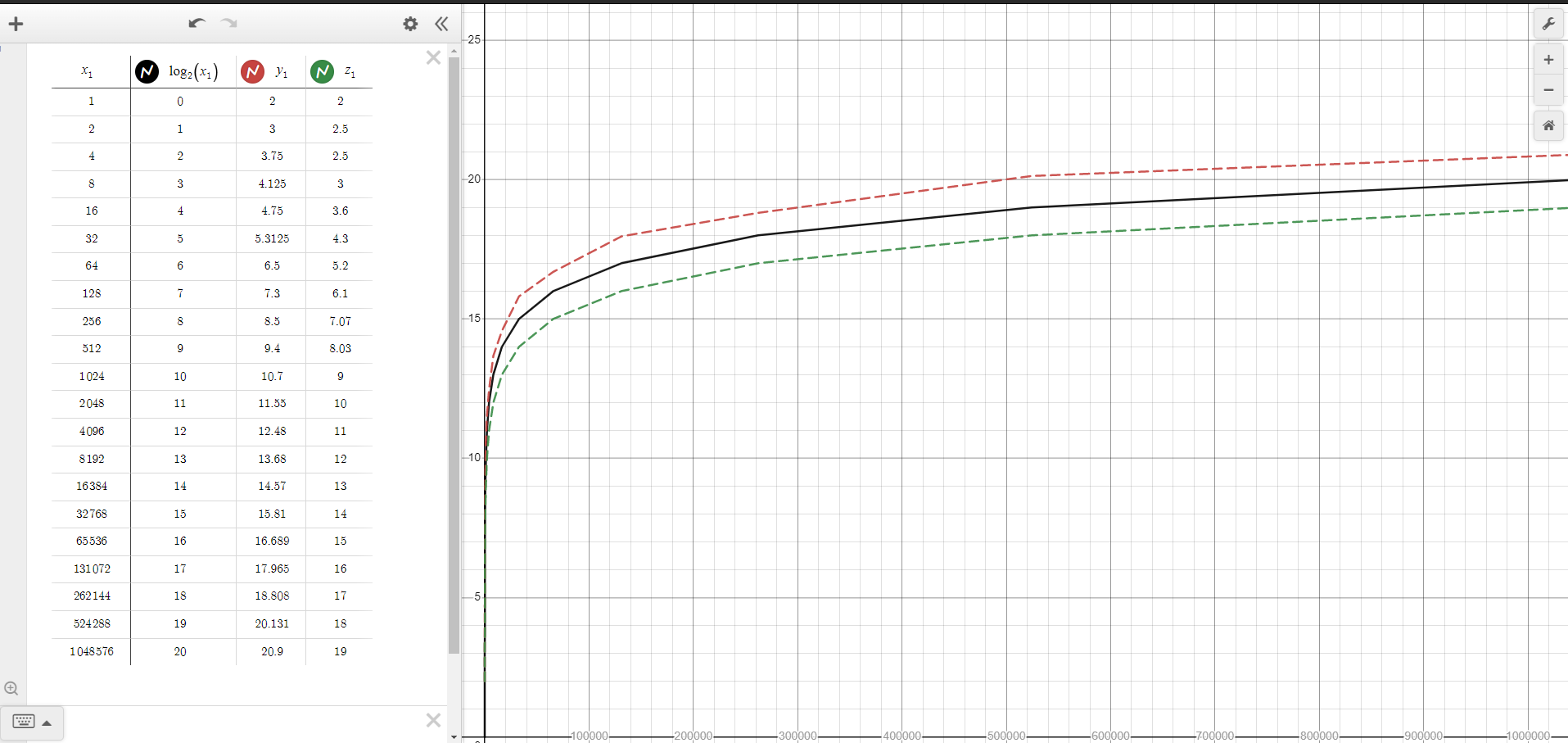
**Search Algorithms Write-Up**

Binary Search:

Binary Search thankfully was not too difficult to implement. The pseudo-code provided by the book was easy to understand and easy to translate to Java. The greatest difficulty was in keeping track of which cases required the removal of which half of a given array. I wrote a debug program that kept track of how many times the search returned that the value wasn’t present and that made it very clear which implementation was right and which implementation was wrong.

Fibonacci Search:

Similarly to the Binary Search Algorithm, the provided code was more than sufficient for testing out the Fibonacci Search and analyzing the results. I only needed to implement counting and then the program was done. Below is a graph representing the output of both the Binary and Fibonacci Search.

(The black line shows log2(x), green Binary Search, and red Fibonacci Search)

From the shape of both algorithms’ graphs, we can see that these algorithms will not take any shape other than log2N. This is the expected output and demonstrates that we have implemented the search algorithms correctly. Conversely, this graph can be used to confirm that log2N is the theoretical limit for both of these algorithms as we received pseudo and source code that we believed to be correct for both.