LAB REPORT : COMPSCI 2XB3

LAB SECTION – L02

Teaching Assistant - *Seyed Parsa Tayefeh Morsal*

Submitted By –

GROUP NUMBER : 2

Name : Adhya Goel

Student Number : 400280182

McMaster email : goela10@mcmaster.ca

Name : Mridul Arora

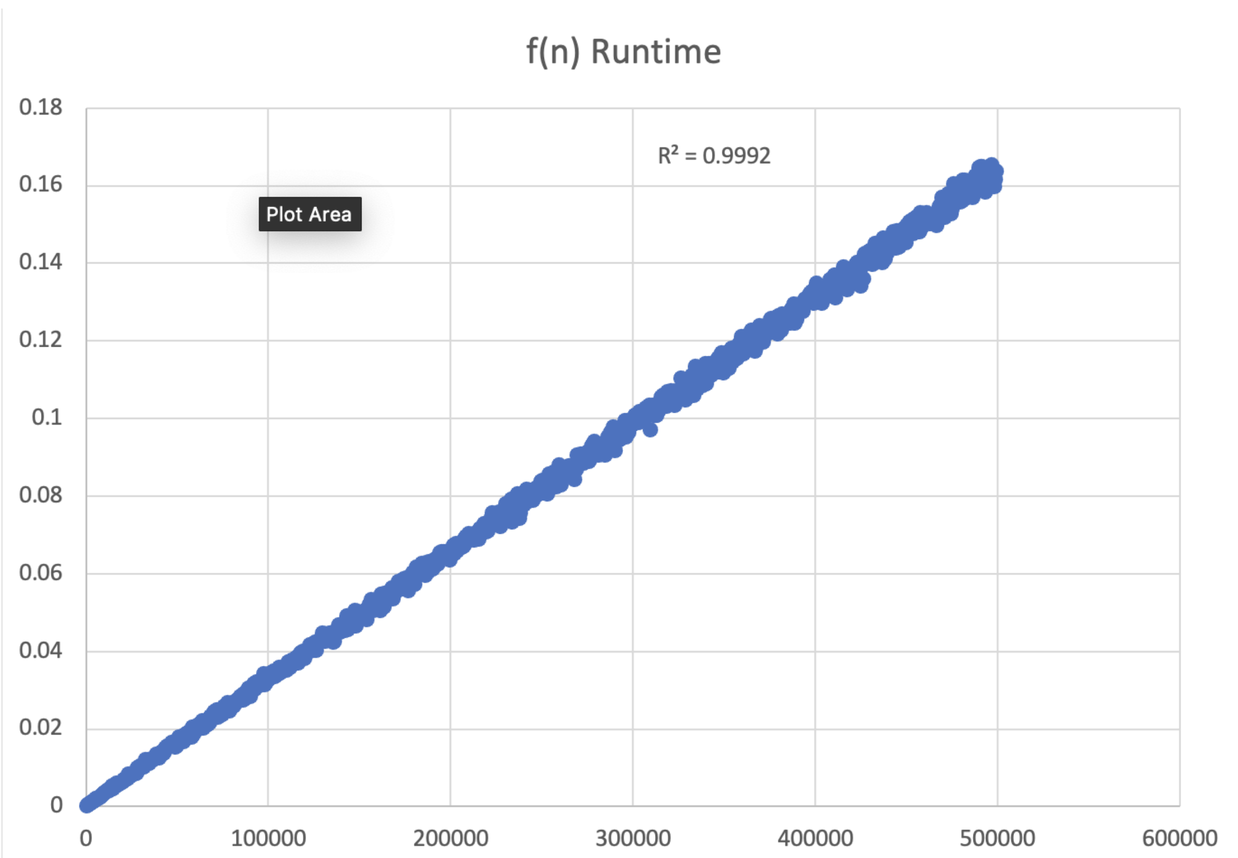
Student Number : 400253526

McMaster email : aroram15@mcmaster.ca

ANALYZE TIMING DATA

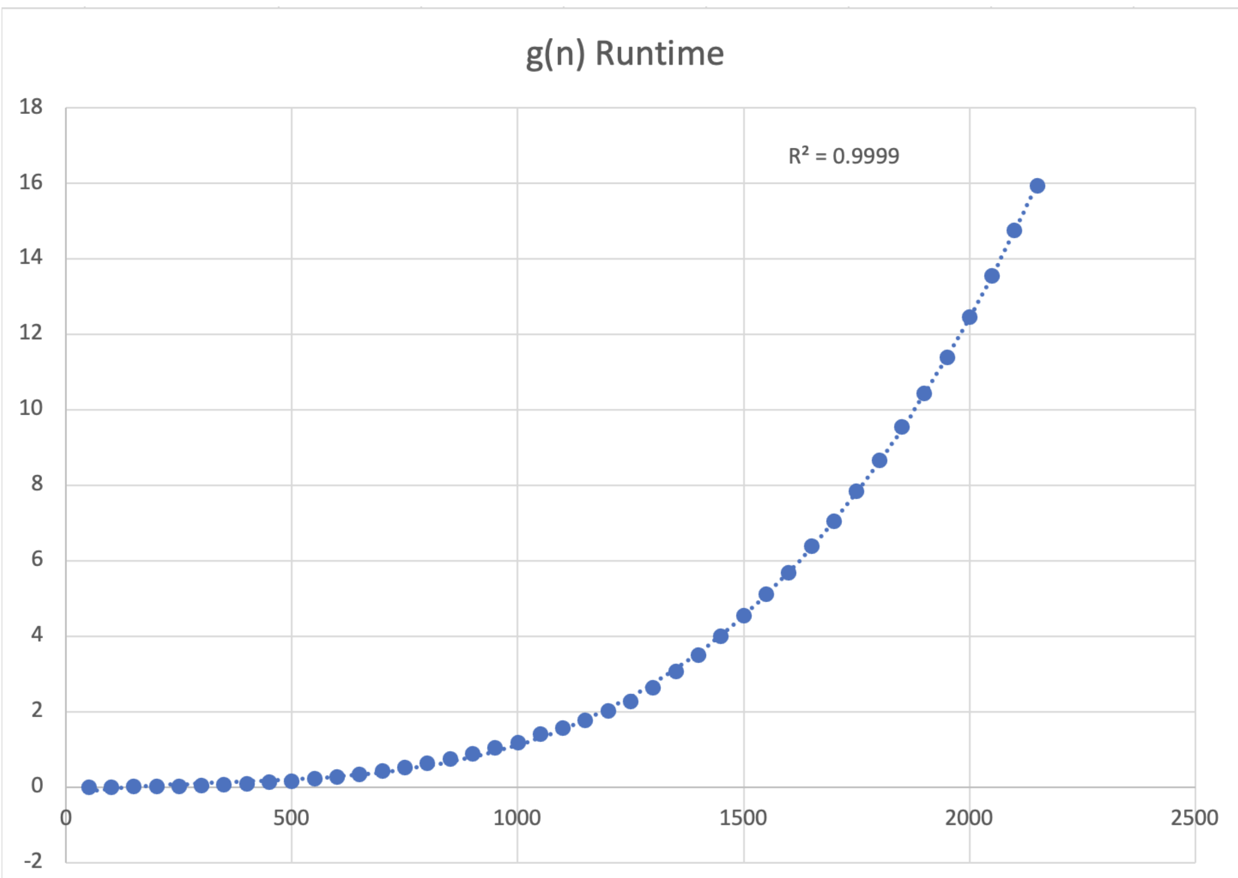
F(n)

According to the graph for this function, the runtime would be linear. As shown in graph, the R sqaured value comes out to be 0.9992 while formatting the trendline. For scientific calculations this number is believed to be very strong indication that the graph actually is linear. Hence, F(n) takes linear time to execute.



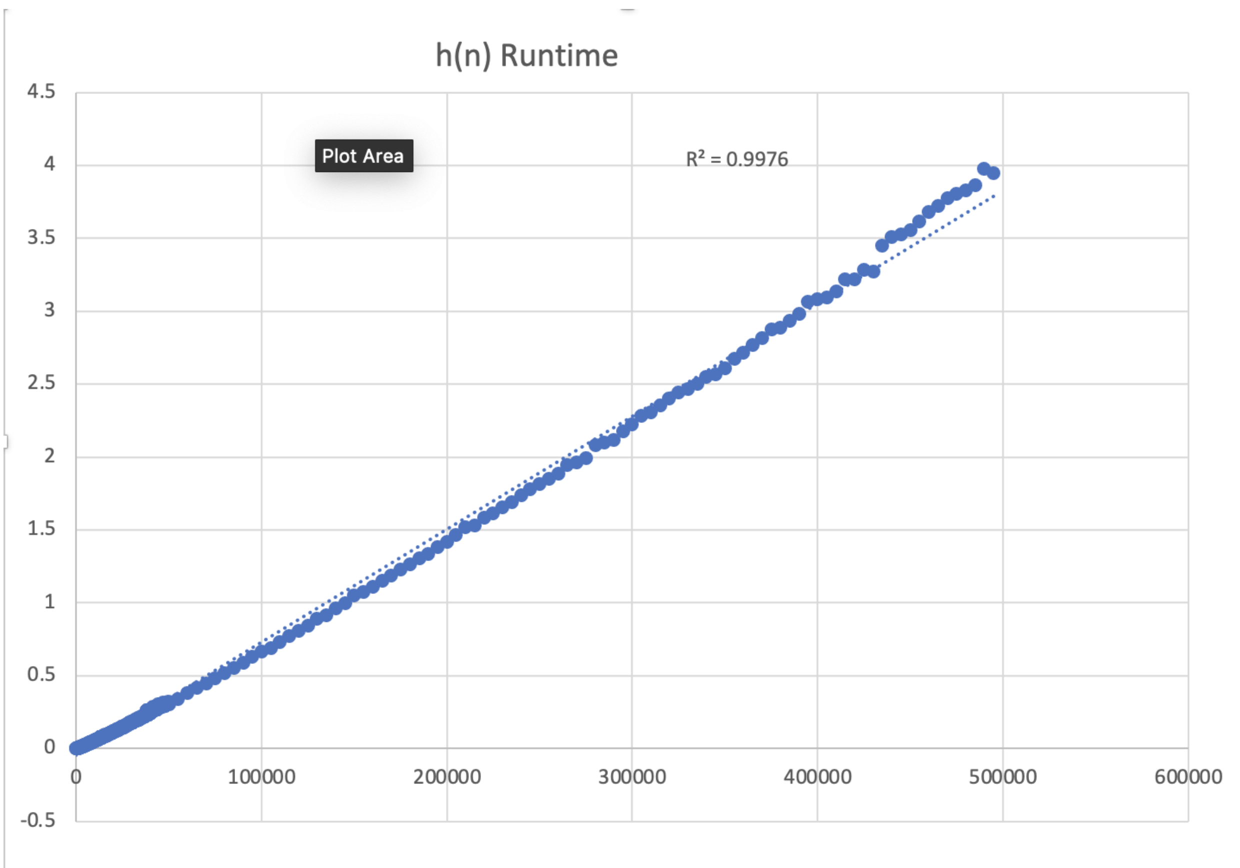
G(n)

For this function the R squared value comes out to be 0.9999 when the trendline is formatted for a polynomial of degree 3 as shown in the graph. The R squared value is strongly suggesting that this graph closely resembles a polynomial of degree 3. Hence, G(n) takes n^3 time to get executed.



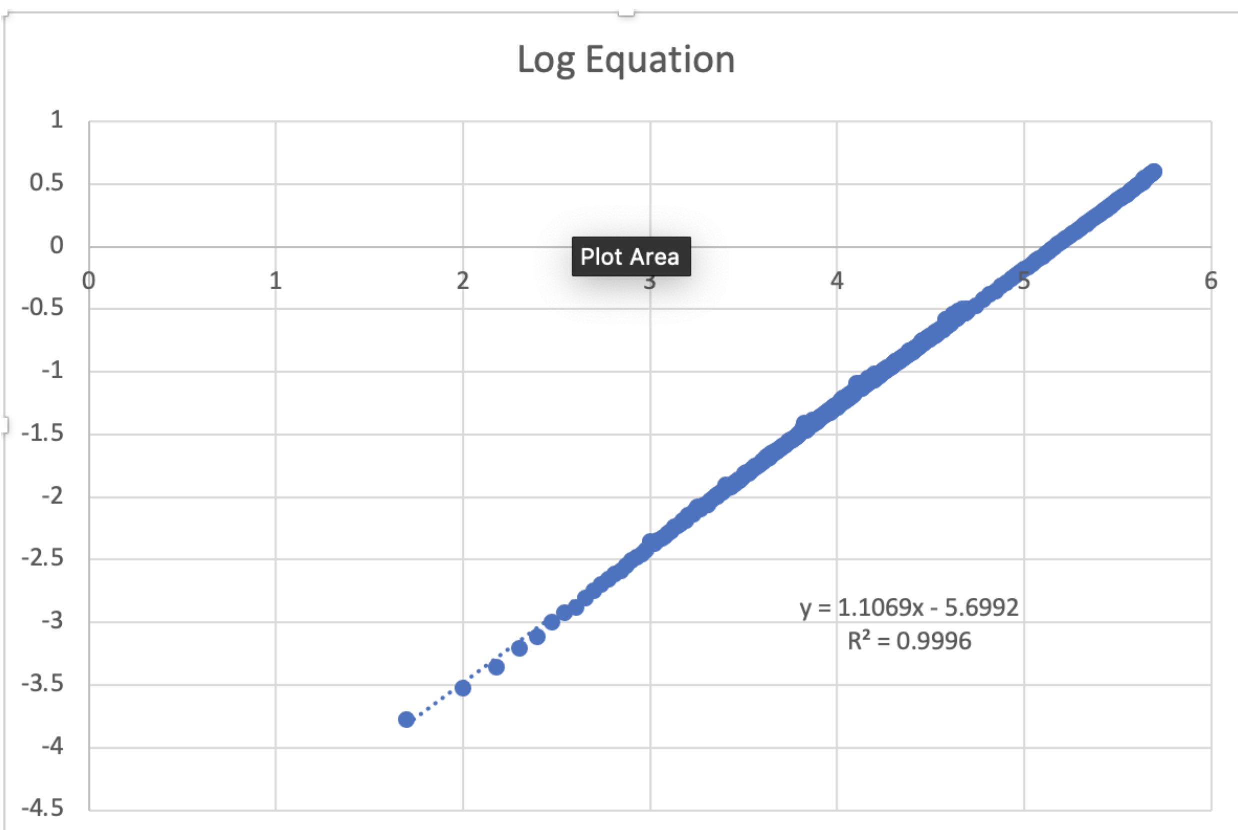
H(n)

This function was a bit tricky as compared to the above 2. The R squared value for this function comes out to be 0.9998 and 0.9999 when the trendline is formatted to be a polynomial of degree 3 and polynomial of degree 4 respectively. Also, when the trendline is formatted to be linear, the R squared value comes out to be 0.9976. This information could have mislead anybody and the person would have assumed that this function resembles a polynomial of degree 4. But the truth is that this function actually is linear.



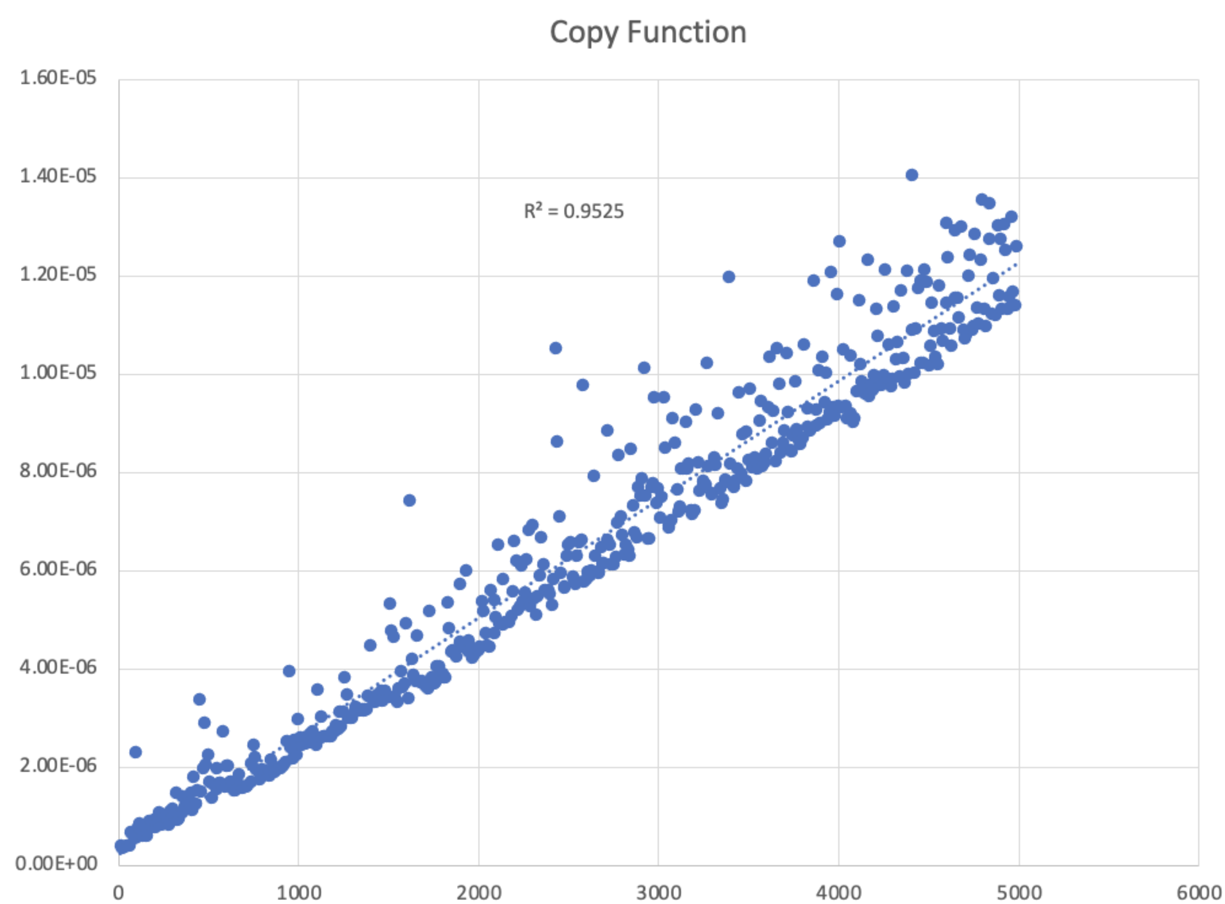
The confusion can be solved by plotting the data points as the logarithmic function of each side of the equation y = cn^k, which comes out to be

log (y) = log (c) + k * log (n). After plotting log data points on the graph, look at the slope of the function. The slope of the function will give the power of the polynomial function which H(n) resembles. Clearly in the graph below, the slope of the function is 1.1069, which says that the fucntion H(n) resembles the polynomial function with power 1, which ultimately is a linear function. Hence, H(n) resembles a linear function.

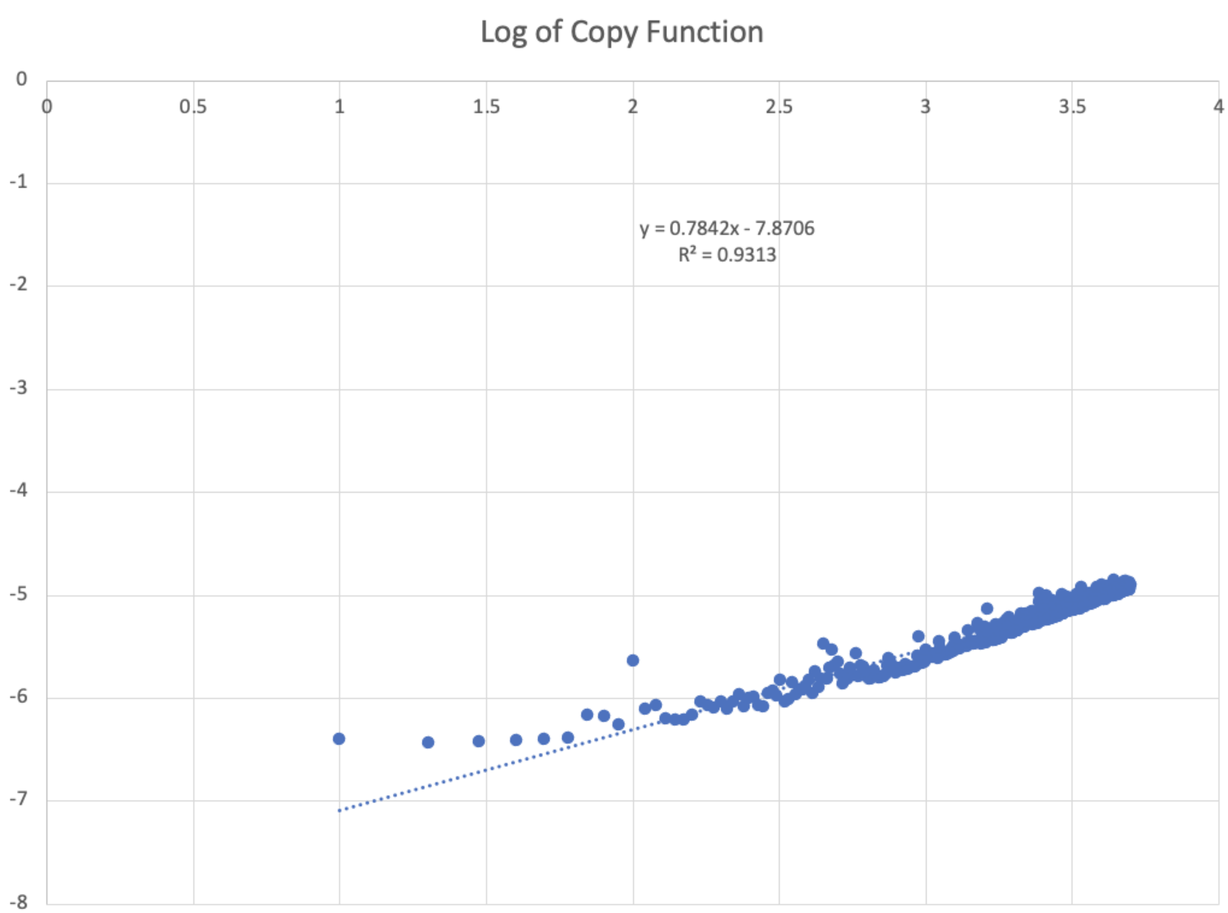


Copy Function (Python List)

For the testing part of copy function, the output of the test will be input size followed by the average time it takes to run the copy function on that input size. When the input gradually increases, the time function takes also increases. On observing the behaviour of the graph, it looks like the graph is linear and indeed it is.

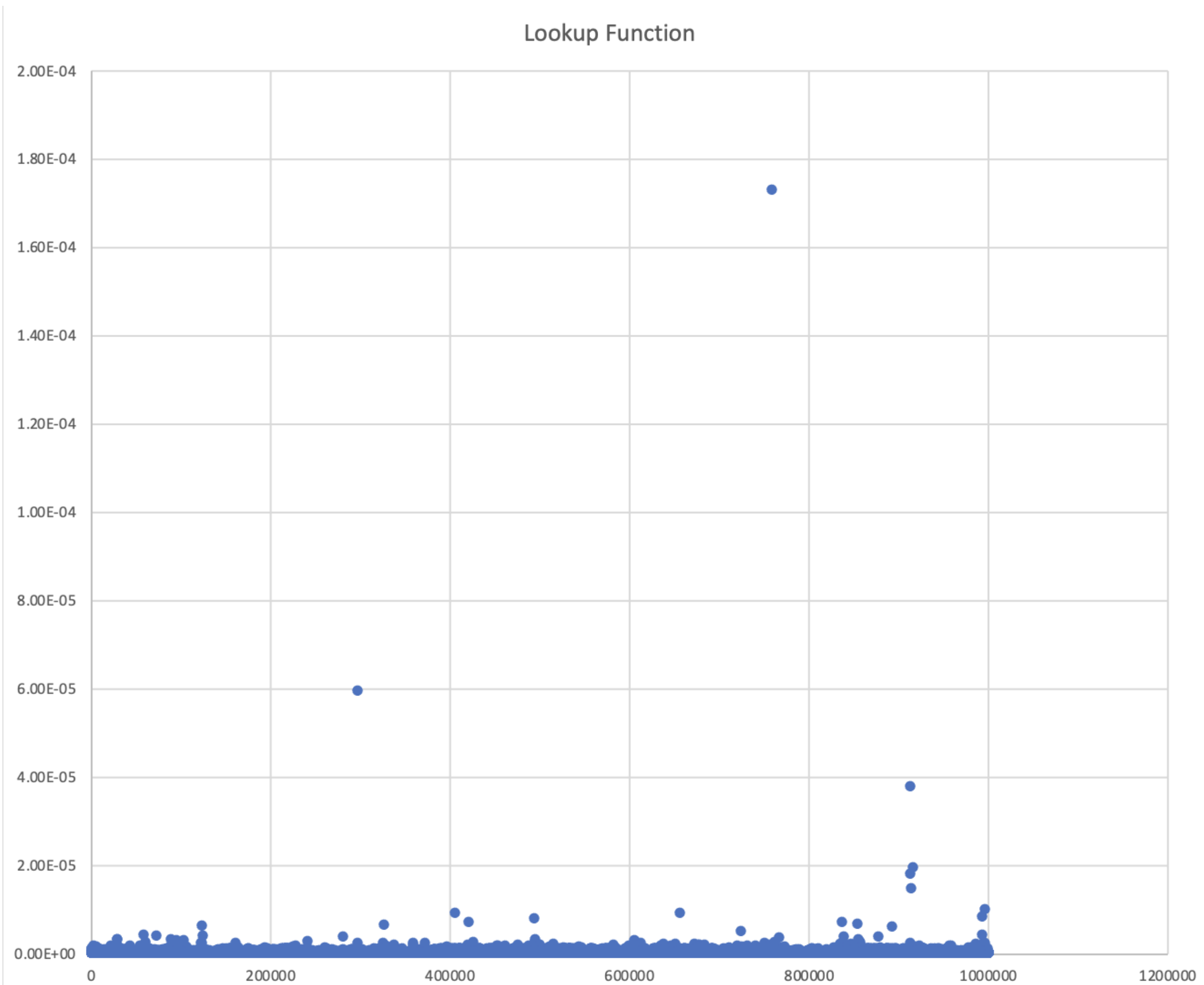


Taking a look only at R squared value doesn't guarantee you the kind of graph it could be, so aside from R squared value individual should also plot log graph function, the same way done for H(n). The result for the slope of the function came out to be ~ 1, which made it clear that the copy function take linear time to execute. The main reason for copy to behave in such a way is that to copy all the elements in a list to another list, you have to iterate over whole list. So, for instance if your list has n values then assuming each iteration takes 1 second, the copy function will be executed in n seconds. This shows that the input is directly (linearly) related to time the function will take.

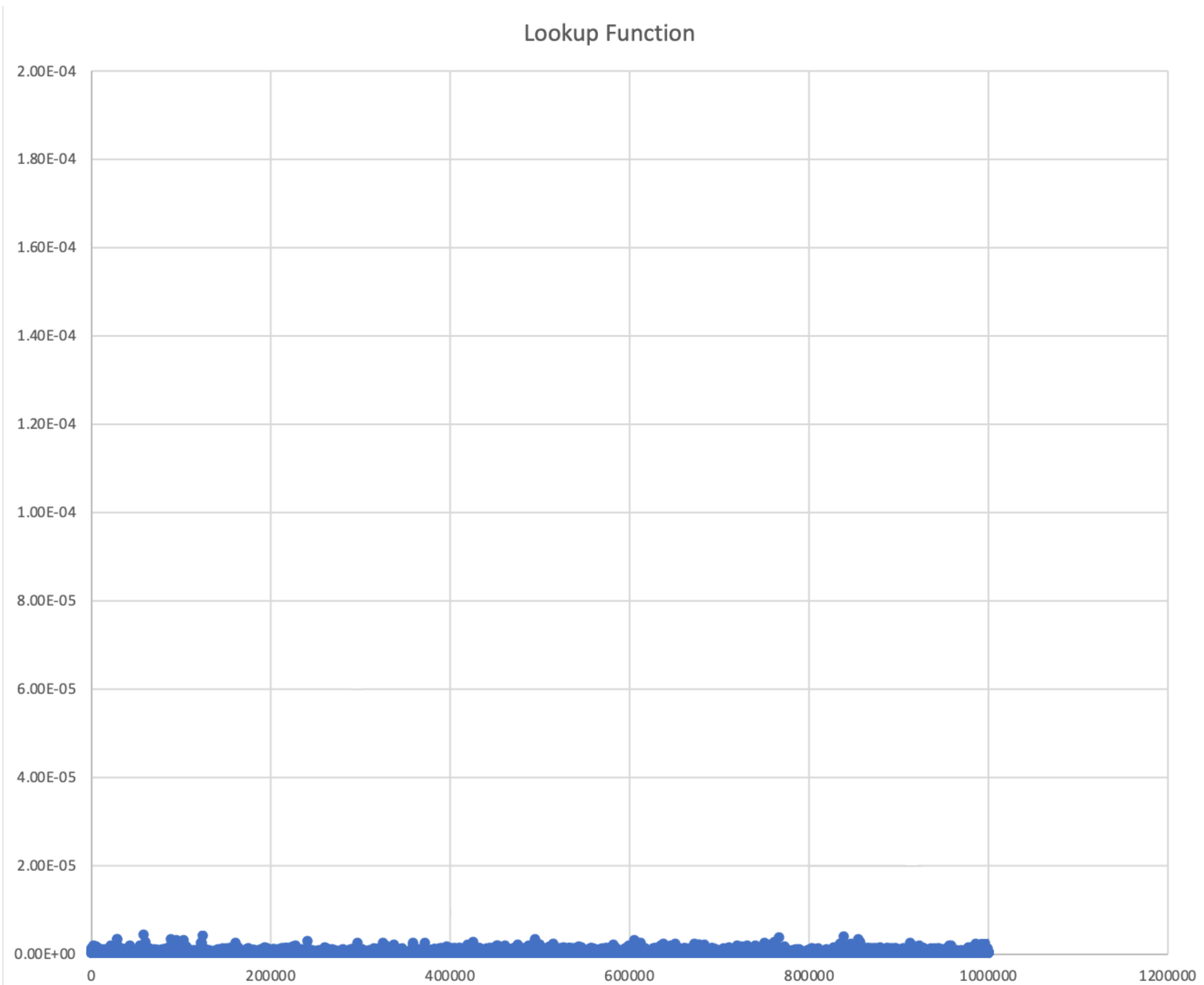


LookUp Function (Python List)

From previous knowledge, the runtime for this particular function should be a constant. After generating the data points and running the experiment, it seems that the previous knowledge was correct.

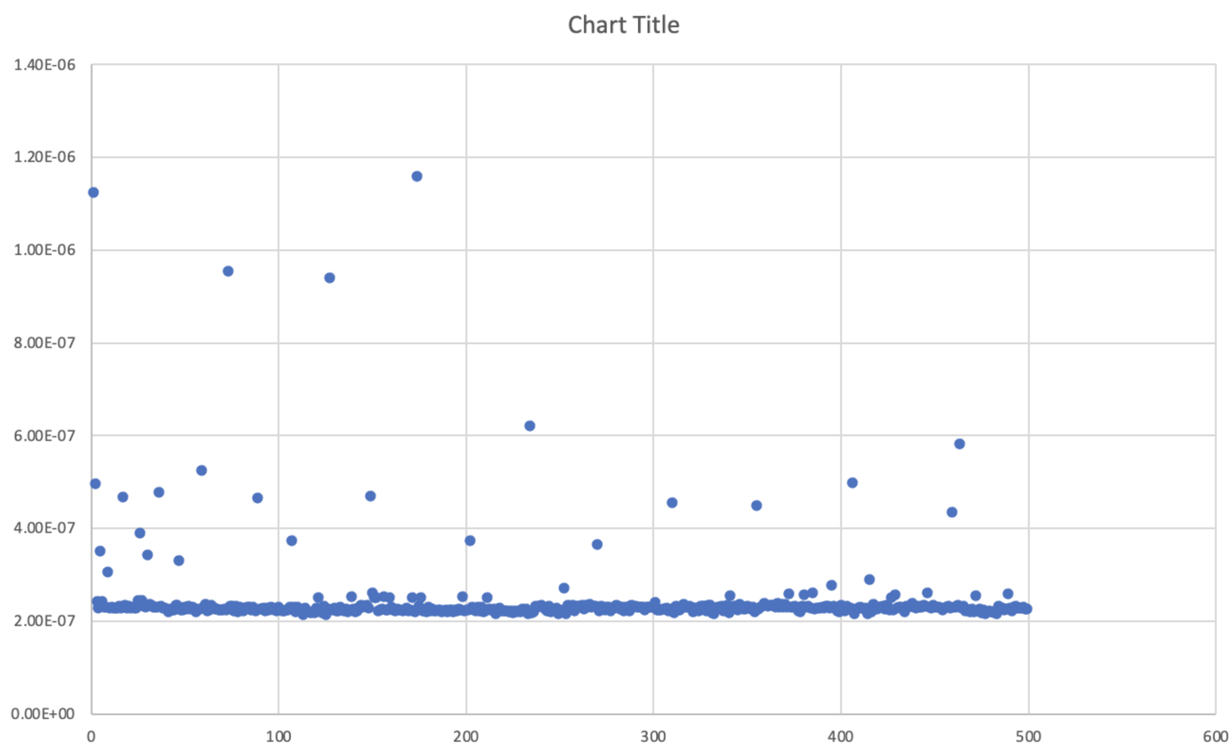


Potential problems in the experiment include random data points which get generated due to background processes. One way to fix this issue would be to minimize the background processes. The other possible method to get accurate data would be to increase the number of runs so that the average gets closer to the actual number. The scattered plot can be seen below after minimizing the background processes.



Append Function (Python List)

Similarly, as the lookup function above the append function should also run with constant time. After analyzing the data points, it seems that the prediction was correct. As before, potential problems include abnormal spikes in runtime which lead to ambiguous data points that are not relevant to the experiment.



Further proceeding, earlier data for the list of length 500, the experiment was now performed on the list of length 10,000. As before, the prediction remains the same for this experiment as well. After determining the graph it shows that the runtime of the function is constant.