Chia Hsuan Wu
42764118
y4d8

Question 2b:

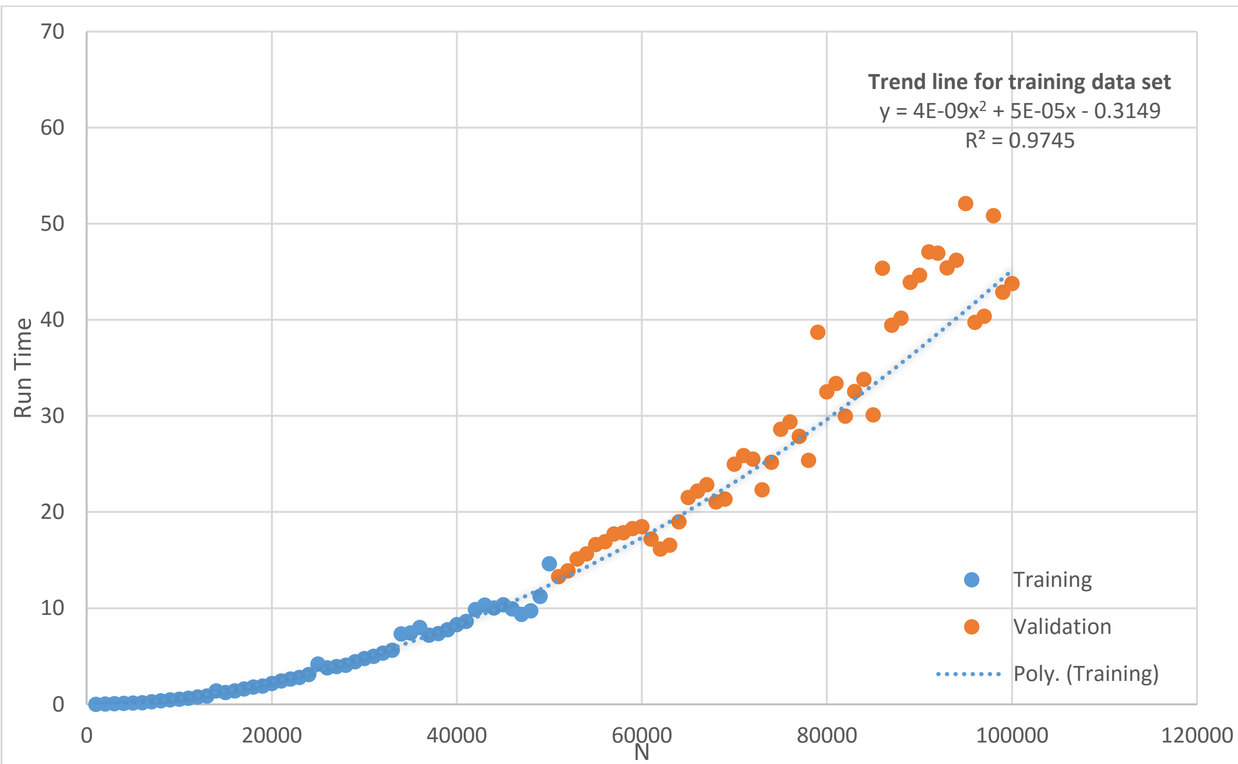
Does the run-time appear to be linear, N log N, quadratic, exponential, or something else (if so, what)? Justify your answer based on your data reported for question2a.

I modified the run time measurement code from mini assignment 1 so that it works with the code in Q2a. I measure the run time 3 times and all 3 run results seem to be very consistent with each other.

This is the average result:

For actual data: please refer to "Q2a actual data.pdf"

After running the measurement from N = 1000 to 100000 with an interval of 1000. I divided the data into 2 parts, training data set and validation data set. The rationale is that, if the function is indeed quadratic, then the trend line formula calculated by Excel on the training data set should be able to predict the testing data sets with relatively high degree of accuracy.



The trend line generated by excel on the training data set is indeed a very good approximation of the validation data. This makes me believe that the running time is indeed quadratic!

Chia Hsuan Wu
42764118
y4d8

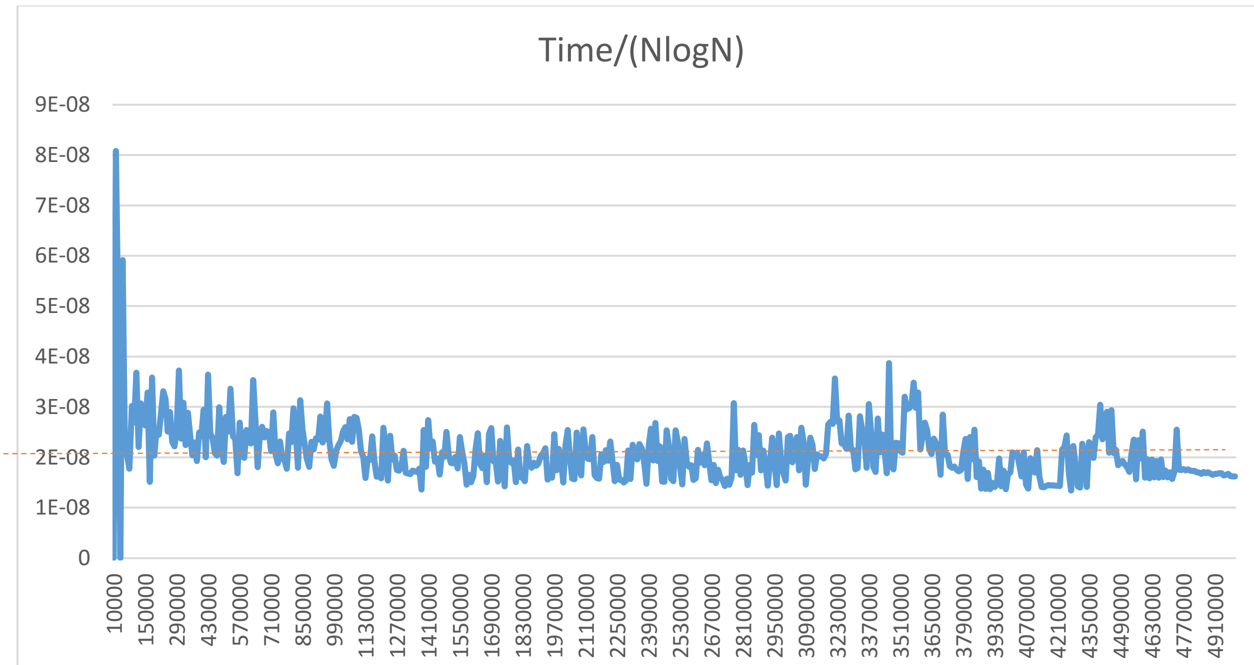
Question 2d:

Experimentally verify the O(N log N) run time.

The performance of the custom made subtract function in Q2c is noticeably faster than lists:subtract.

I ran the function from N = 10000 to 5000000 with interval of 10000 and then measured the run time with the monotonic_time() function similar to mini assignment 1. Then, I calculated the value of N log N for each N value. Finally, I divided the run time by N log N and plot it on to a graph. (This experiment method is suggested by Mark on Piazza).

For actual data: please refer to "Q2d actual data.pdf"



The ratio stayed relatively constant throughout (despite the high degree of fluctuation in the very beginning). Since the k value for k(N log N) stayed about constant even at high N value, it gives us a very strong confidence that the run time for the custom made function is indeed O(N log N).