Part 3 – Performance Testing Report

URL Links:

Files used to run these performance tests:

Source Code: https://github.com/rpalival/SSW567A-Final-Project-Group-9/blob/main/encode.py

Source Code: https://github.com/rpalival/SSW567A-Final-Project-Group-9/blob/main/decode.py

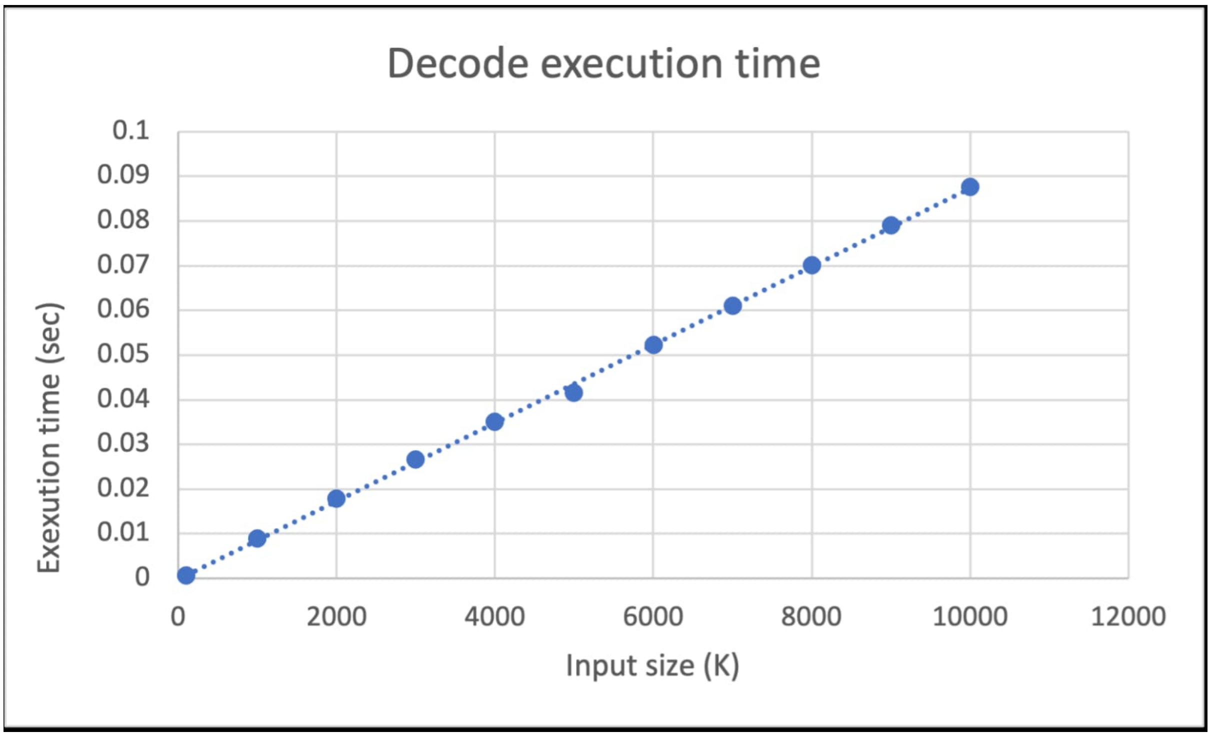
Raw Data in CSV

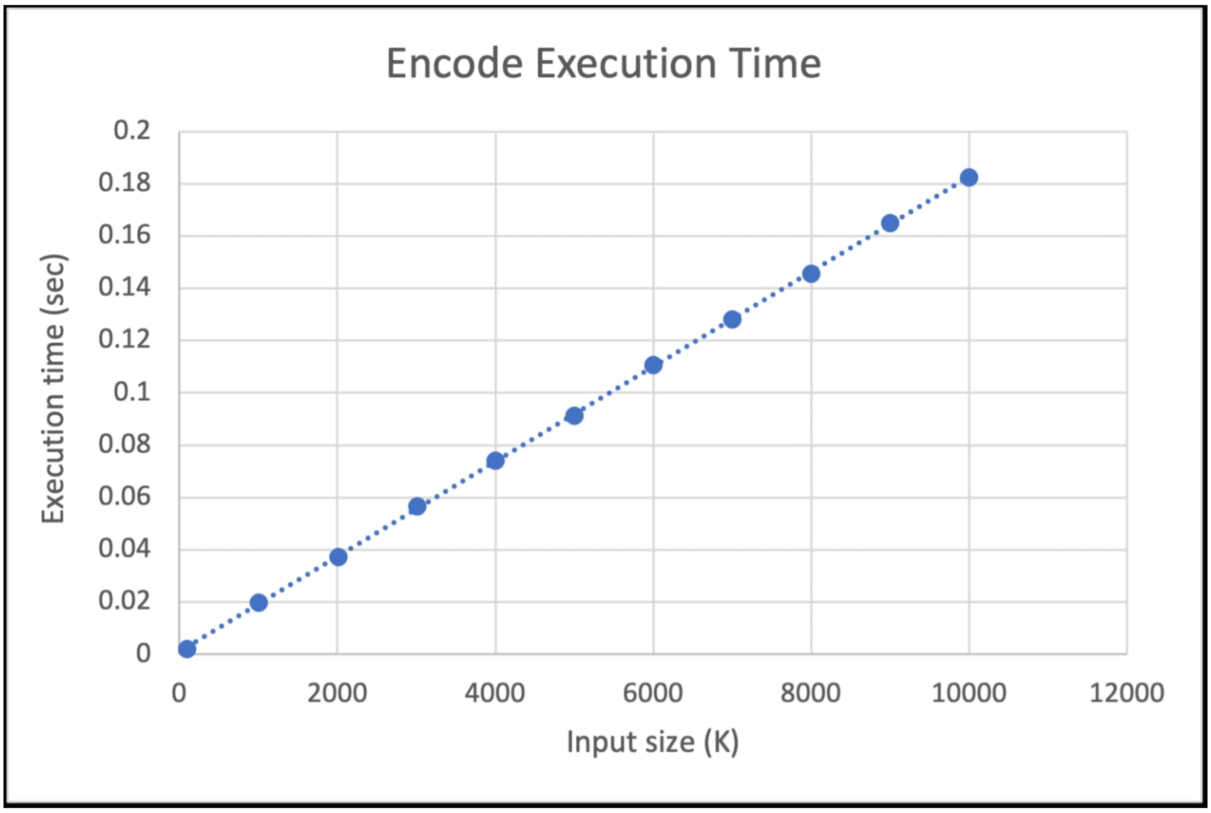
File 1: https://github.com/rpalival/SSW567A-Final-Project-Group-9/blob/main/decode_execution_time.csv

File 2: https://github.com/rpalival/SSW567A-Final-Project-Group-9/blob/main/encode_execution_time.csv

Functionality:

- In part-3 we conduct performance testing on unit test cases that has been conducted in part 2.
- The task is to measure the execution times to process the first 100 records, the first 1000 records, and the first *n* thousand records for n running from 1 to 10.
- We input decode and encode function to measure its performance with different input sizes.
- We will use the Python timing libraries to measure how long your program takes to run without test cases and with test cases.
- We Install timing library to Read json file and input the element as string into the encode and decode function as well as timing library .
- we take a string from the hardware scanner and perform Decode function . We parse the encoded json file into the decode function and Store the string in the information field. Every time we pass the string it get splited and timed.
- In encode function each input is basically a dictionary. We pass decode json file into the encode function which will return the file line as the result we have obtained here, along with it we are giving the four field to compute the check digit of the number.





- As it can be seen in both the graphs are linear and explains that the execution time did not change drastically for any K number of tests or so. Hence the performance was consistent.
- The Actual times are written down in the csv files.