This project is written in Python 3.7.4.

Task 1

This program generates a 16 byte, 1 kilobyte, and 3 kilobyte file. It then hashes the contents of the file for 1 second and outputs the number of hashes. The first hashing algorithm used is SHA256. Here are the results, first 3 are with SHA256.

* 16B hashed 1842 times in 1 second.
* 1kB hashed 1122 times in 1 second.
* 3kB hashed 1488 times in 1 second.

Next 3 are hashed with BLAKE2b

* 16B hashed 2067 times in 1 second.
* 1kB hashed 1179 times in 1 second.
* 3kB hashed 1479 times in 1 second.

Hardware Specs

* i5-6600k overclocked at 4.5Ghz
* 16GB 3000Mhz Memory
* 250GB Samsung 850 evo SSD

Initially I was surprised that the 3kB file was faster than the 1kB. I expected performance to decrease at least a little bit. I figured I would see what would happen if I increased the file size of the 3kB by 10x and there was a significant decrease in performance. Then I increased it’s size by 10x again, and once again there was a substantial decrease in performance. Perhaps I just needed larger file sizes to see the cost of running these algorithms. BLAKE2b performed in the same way. Large dip in performance from 16B to 1kB but an increase in performance at 3kB.

The odds of finding a collision with random input is 1 in 2­255. At 1842 hashes per second I could do ~58 billion hashes a year. Let’s assume I only need to do 2255/2 hashes. My odds would still be 1 out of 2.8948022e+76. It would take roughly 58,000,000,000 \* 1056 years.