Part 1 - Variant of Dining Philosophers with Waiter

5 Chopsticks	
Run 1	6669.87 ms
Run 2	6673.09 ms
Run 3	6675.72 ms
Run 4	6675.02 ms
Run 5	6674.32 ms
Avg	6673.60 ms

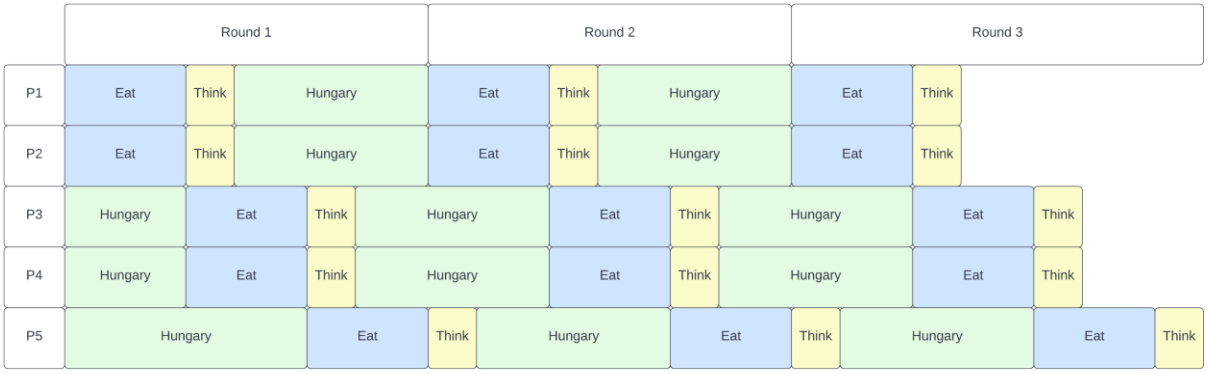
7 Chopsticks	
Run 1	2668.95 ms
Run 2	2666.74 ms
Run 3	2669.14 ms
Run 4	2667.01 ms
Run 5	2669.49 ms
Avg	2668.27 ms

9 Chopsticks	
Run 1	2337.56 ms
Run 2	2333.99 ms
Run 3	2334.83 ms
Run 4	2333.41 ms
Run 5	2331.54 ms
Avg	2334.27 ms

6 Chopsticks	
Run 1	2666.59 ms
Run 2	2671.28 ms
Run 3	2668.32 ms
Run 4	2666.59 ms
Run 5	2669.49 ms
Avg	2668.45 ms

8 Chopsticks	
Run 1	2334.11 ms
Run 2	2333.40 ms
Run 3	2334.77 ms
Run 4	2332.84 ms
Run 5	2333.95 ms
Avg	2333.81 ms

10 Chopsticks	
Run 1	0.12 ms
Run 2	0.12 ms
Run 3	0.11 ms
Run 4	0.13 ms
Run 5	0.12 ms
Avg	0.12 ms



Theoretical average wait time = (5 + 5 + 10 + 8 * 10) / 15 = 6.66 seconds

Part 2 - Composite Numbers & Multithreading

composite
76.8176 Milliseconds
real 0m0.106s
user 0m0.078s
sys 0m0.015s

composite threaded - 2
49.283 Milliseconds
real 0m0.083s
user 0m0.108s
sys 0m0.031s

composite threaded - 8
24.0004 Milliseconds
real 0m0.053s
user 0m0.062s
sys 0m0.030s

composite threaded - 1
78.1234 Milliseconds
real 0m0.109s
user 0m0.093s
sys 0m0.015s

composite threaded - 4
26.7145 Milliseconds
real 0m0.056s
user 0m0.109s
sys 0m0.015s

composite threaded - 16
23.9606 Milliseconds
real 0m0.053s
user 0m0.093s
sys 0m0.046s

As we can see using more threads lowers the running time of the program. As we start to increase the number of threads the computational time reaches a stable number thus using more threads won't make the program run faster. It can be also said that increasing the number of threads after a certain point will lead to increasing the computational time.

The system time increases as we increase the number of threads this reflects the parallel workload of this program. This also shows that creating more threaded takes CPU time and will lead to diminishing returns after a point.