

TESTING

The efficiency of our implementation was tested exclusively on the input data subset file "input30m" (Size: 30mb) as it provided reasonable execution times for both the single and multi-thread versions of the program.

BEFORE PARALLELISM

We quickly realized by using a larger input file than the one we used in the first two assignments, that our program was flawed. It utilized a few but very crucial $O(n)$ operations that would slow down the execution to the point of never really ending. So our first goal before multi-threading was removing all of these functions and adding Hash Tables instead to produce the same result but with $O(1)$ complexity. This important change caused our program to produce the correct results in just 54m, a good execution time that would allow us to contrast the efficiency of our program later on.

AFTER PARALLELISM

Execution times

Machine 1 [CPU: AMD RYZEN 5 5600X 6-Core Processor RAM: 16GB]

Threads	Time
2	1591641[26m:31s:641ms]
4	931954[15m:31s:954ms]
8	1302835[21m:42s:835ms]
16	668436[11m:8s:436ms]
32	645272[10m:45s:272ms]
64	619453[10m:19s:453ms]
128	627010[10m:27s:10ms]

Machine 2 [CPU: Intel Core i7 6700K RAM: 16GB]

Threads	Time
2	2324189[38m:44s:189ms]
4	1483935[24m:43s:935ms]
8	1700657[28m:20s:657ms]
16	2202135[36m:42s:135ms]
32	1730668[28m:50s:668ms]
64	1819608[30m:19s:608ms]
128	1446055[24m:6s:55ms]