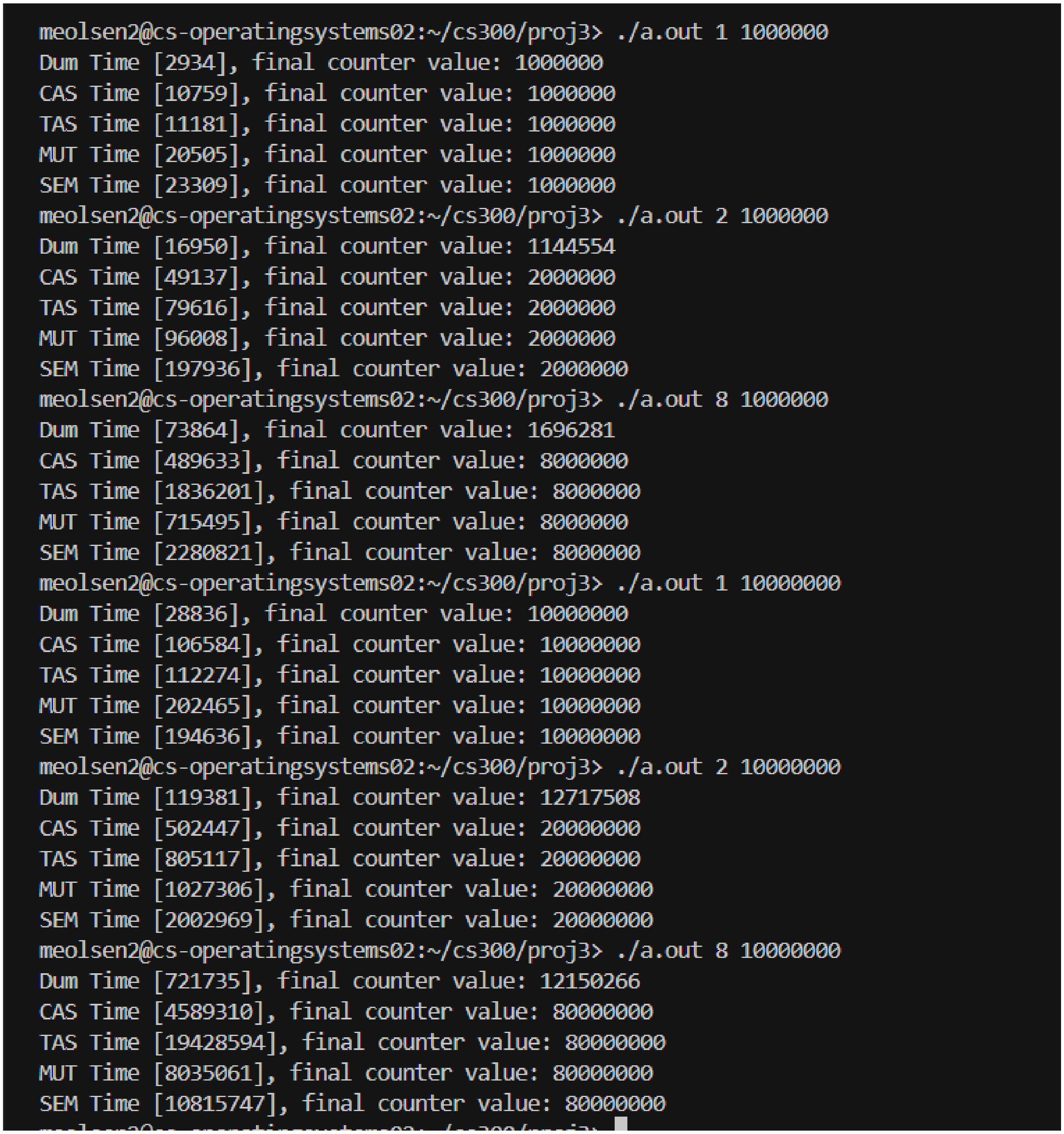
I. Outputs



```
meolsen2@cs-operatingsystems02:~/cs300/proj3> ./a.out 1 1000000
Dum Time [2934], final counter value: 1000000
CAS Time [10759], final counter value: 1000000
TAS Time [11181], final counter value: 1000000
MUT Time [20505], final counter value: 1000000
SEM Time [23309], final counter value: 1000000
meolsen2@cs-operatingsystems02:~/cs300/proj3> ./a.out 2 1000000
Dum Time [16950], final counter value: 1144554
CAS Time [49137], final counter value: 2000000
TAS Time [79616], final counter value: 2000000
MUT Time [96008], final counter value: 2000000
SEM Time [197936], final counter value: 2000000
meolsen2@cs-operatingsystems02:~/cs300/proj3> ./a.out 8 1000000
Dum Time [73864], final counter value: 1696281
CAS Time [489633], final counter value: 8000000
TAS Time [1836201], final counter value: 8000000
MUT Time [715495], final counter value: 8000000
SEM Time [2280821], final counter value: 8000000
meolsen2@cs-operatingsystems02:~/cs300/proj3> ./a.out 1 10000000
Dum Time [28836], final counter value: 10000000
CAS Time [106584], final counter value: 10000000
TAS Time [112274], final counter value: 10000000
MUT Time [202465], final counter value: 10000000
SEM Time [194636], final counter value: 10000000
meolsen2@cs-operatingsystems02:~/cs300/proj3> ./a.out 2 10000000
Dum Time [119381], final counter value: 12717508
CAS Time [502447], final counter value: 20000000
TAS Time [805117], final counter value: 20000000
MUT Time [1027306], final counter value: 20000000
SEM Time [2002969], final counter value: 20000000
meolsen2@cs-operatingsystems02:~/cs300/proj3> ./a.out 8 10000000
Dum Time [721735], final counter value: 12150266
CAS Time [4589310], final counter value: 80000000
TAS Time [19428594], final counter value: 80000000
MUT Time [8035061], final counter value: 80000000
SEM Time [10815747], final counter value: 80000000
meolsen2@cs-operatingsystems02:~/cs300/proj3>
```

II. Analysis

a. What are the final counter values with Dum counter? Explain.

When there are multiple threads, counter_dum is less than maxcount in every test case that I have tried. Once there are multiple threads, the Dumworker function can't stop a race condition from happening, since there's no synchronization tools used in this method. This means that the critical section is unprotected, and counter_dum isn't guaranteed to be accurate.

b. How NumThread affects the execution time of the different implementations?

Each implementation's execution time increases with an increase in num_threads. A higher number of threads means that (a) more threads need to be created and joined, which takes time and (b) more threads have to count to maxcount. Each thread counts to maxcount, so a 2-thread test case for maxcount = 1000 would result in a final count of 2000, whereas a 4-thread test case would result in a final count of 4000.