REPORT

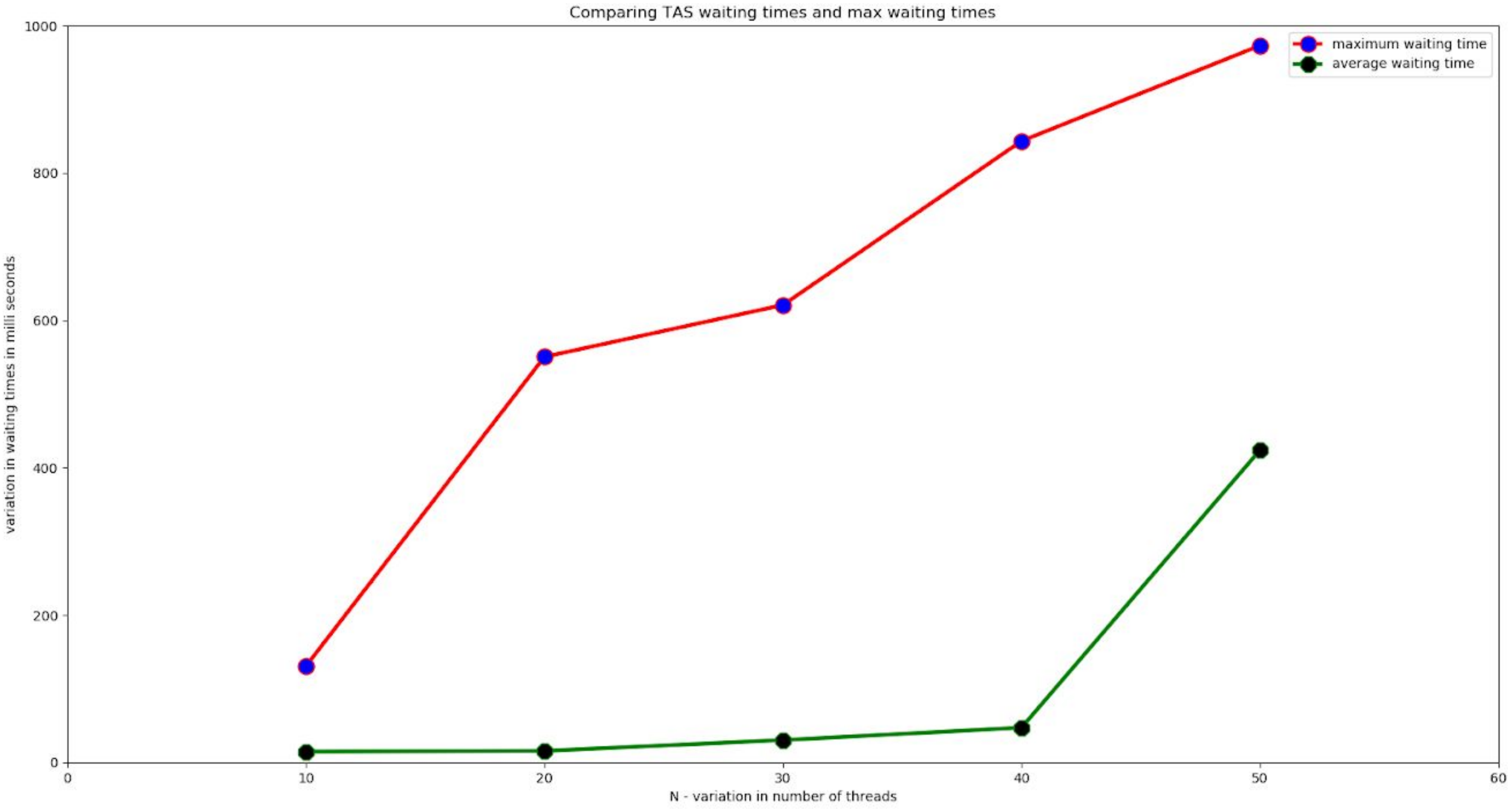
SRI HARI. M
CS19BTECH 11039

Aim: Comparison between the Mutually exclusion algorithms

Codes design:

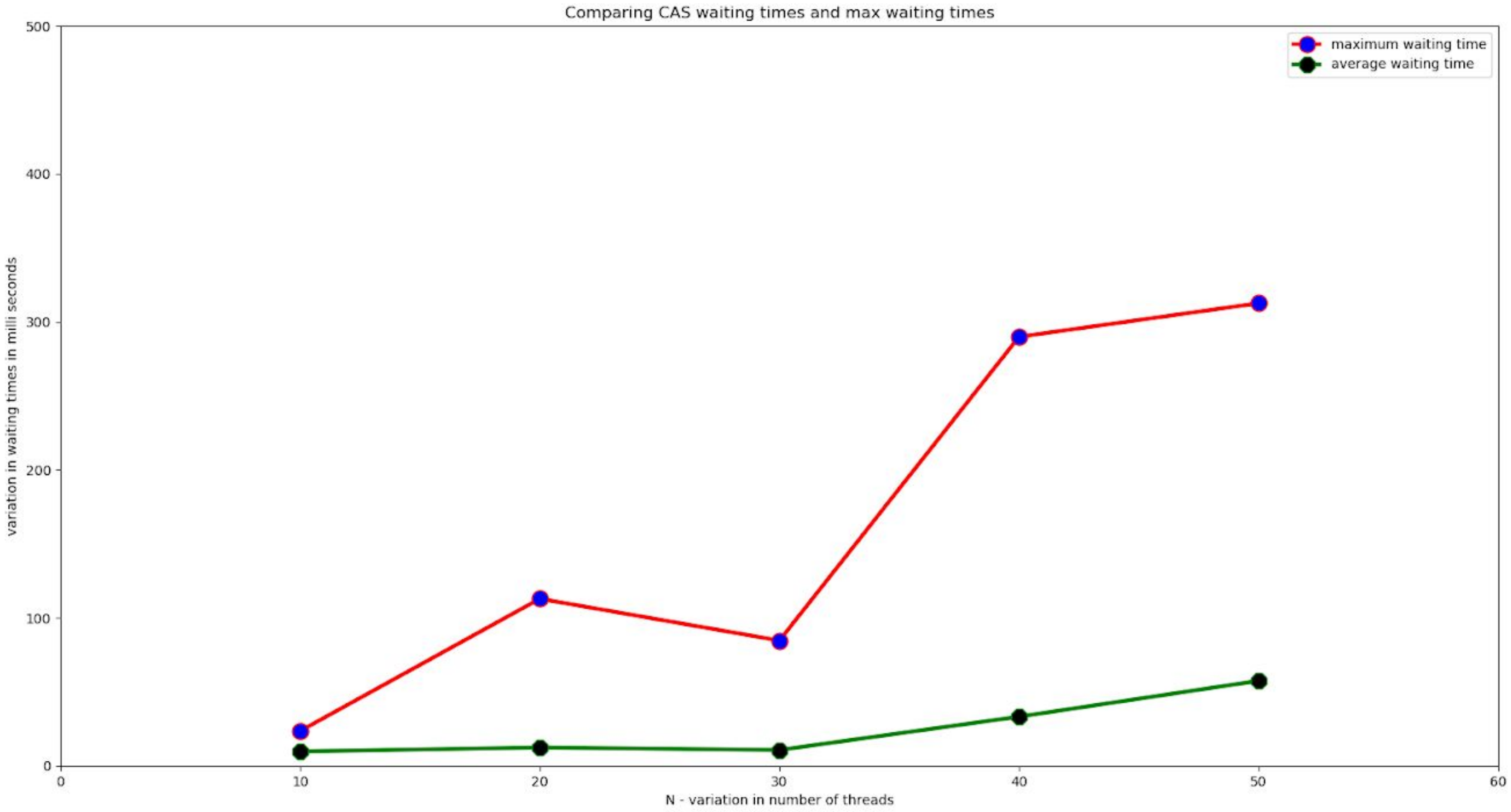
Given the template in the problem statement, it is used for the test critical section function. Initially, after including all the required header files, worker threads are created of the desired size after fetching the inputs from the file “inp-params.txt”. After creating the threads, the function test critical section is called from the main function. The function takes id as input, which will be helpful for printing the thread ids. In **TAS** an atomic flag variable is initialized and the inbuilt test_and_set() method is used in the entry section and clear() is used in the exit section. In **CAS** an atomic bool variable is used and the inbuilt method, atomic_compare_exchange_strong() is used in the entry section and the exit section clears the flag. Whereas in **CAS-Bounded** a boolean array of the size number of threads is initialized and in the entry section it changes the waiting state and uses the CAS method to process and sets back the waiting state to false.

Test and set (TAS):



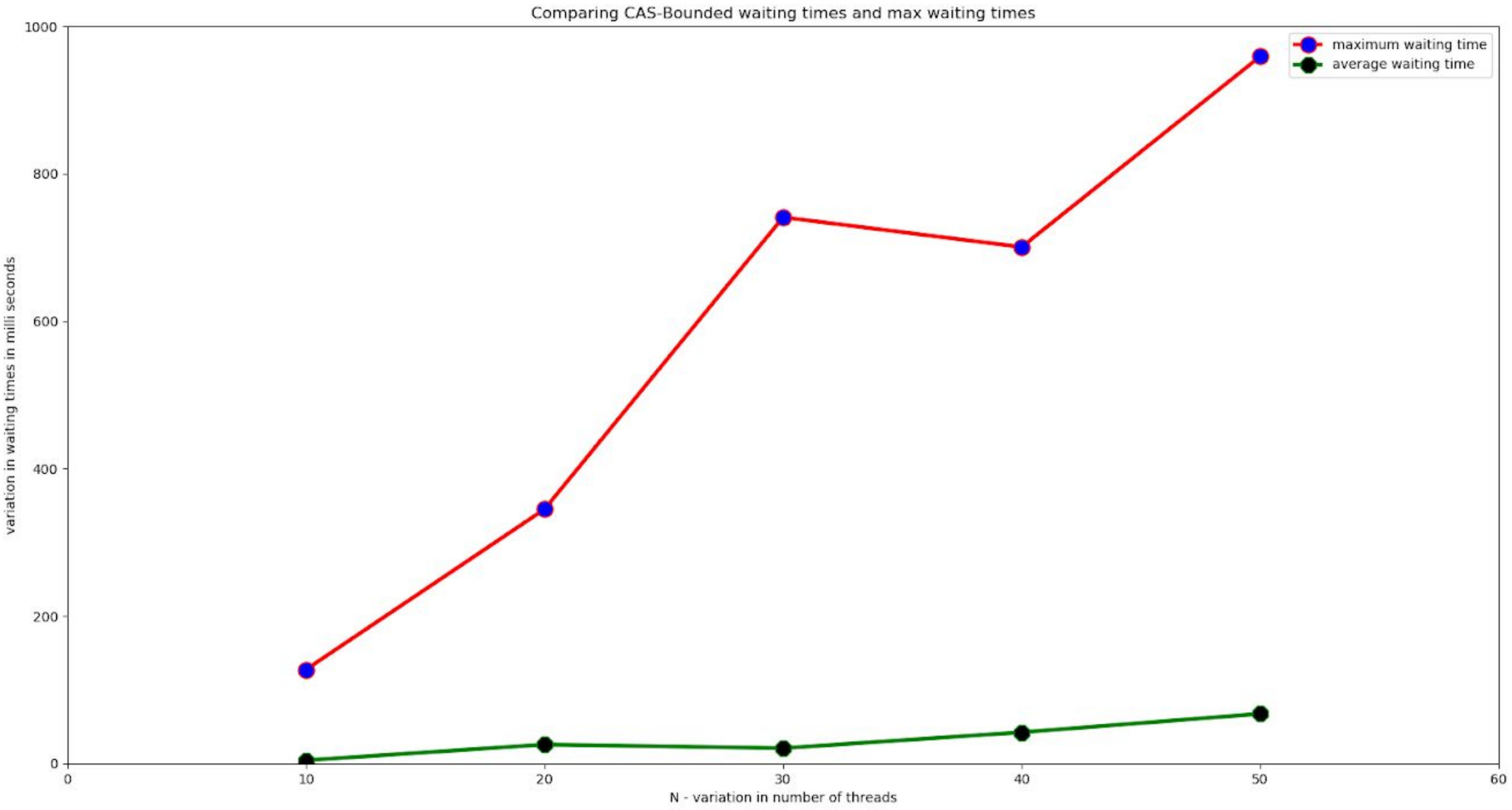
As in the graph, the average waiting time for the algorithm is almost equal to a lesser number of threads. But once the number of threads increases in number, the average waiting time suddenly increases. Where the maximum waiting time is increasing as the number of threads are increasing. So as the sudden rise is seen for the larger number of threads, we can say that there is a chance of **starvation** with a large number of threads.

CAS algorithm:



When compared to TAS algorithm, it has a lesser average waiting time and maximum waiting time. The average waiting time is increasing but less rapid than the TAS algorithm. So here the chance of starvation is less prone than in the case of CAS algorithm.

CAS- Bounded Algorithm:



Here, in this case, the average waiting time is increasing slowly, in a non-rapid fashion. While the maximum waiting times are also not fluctuating at a very high count. Here also the prone of starvation is lesser based on the above graph.

Note:

The values are taken after averaging the values obtained from 10 results. Same with the case with all the algorithms and both the times. The printing in the files is a bit weird as there is no locking system used to print in the file(sir told not to use it).