Assignment 2 – Multithreaded MapReduce Library

Category	Mark
`Functionality` • **Marks will be deducted for:** using FCFS or other policies instead of SJF for the job queue (-5); not keeping the intermediate data structures sorted in ascending key order (-5).	/10 marks
`Correctness` • **Marks will be deducted for:** not passing the public and holdout test cases.	/40 marks
`Runtime` • **Marks will be deducted if:** concurrency management was not done economically or poor decisions were made regarding the use of data structures, as a result taking a huge amount of time to finish large test cases.	/10 marks
`Memory Leaks` • **Marks will be deducted for:** memory leaks (`valgrind` will be used to identify memory leaks).	/10 marks
`Makefile` • **Marks will be deducted for:** missing a required target (-1 each); missing the necessary flags or using the `-O` flag (-3); failing to produce an executable called `wordcount` (-2).	/15 marks
`README` • **Marks will be deducted for:** not listing the synchronization primitives that were used in the implementation of each library and how they were used (-5); missing or incomplete explanation of the data structures used (-3); not describing how the implementation was tested, e.g. using a different number of data splits, threads, or partitions (-2); wrong/incomplete answer to question about why the reduction in running time becomes limited beyond a certain number of threads (-5).	/15 marks

Note: Deductions may be applied at the grader's discretion if the submission:

- Does not meet the requirements mentioned in the assignment description (e.g. no correctness marks will be given if it creates/destroys threads on demand rather than using a thread pool) or has extremely poor code structure or modularity;

- Does not compile or run correctly on a lab machine;

- Does not compile cleanly with `-Wall` (up to 5 marks can be deducted).