

# Proposal – Multi-threaded Search Algorithm

## Team

Group D2

Quinn Murphy (NETID: qmurphy, Section D)

Ben Williams (NETID: bww1, Section D)

## Project Idea

We are going to create an optimized multi-threaded search program that determines different information about a data set. The program at minimum must contain the following:

- Automatic calibration to determine the point at which making threads is no longer effective.
- Be able to calculate the minimum/maximum of the data set and successor/predecessor of an element.
- Sort the data set in  $O(n^2)$
- Compile the algorithm as a library and be able to link it in other programs.
- Extra Credit: Sort the data set in  $O(n\log(n))$
- Extra Credit: Be able to calculate the mean, median, and mode averages and standard deviation of the data set.

## Topics From Class

The following class topics will be included in this project:

- Multi-threading
- POSIX System Calls
- Libraries

## Timeline

Week 1:

- Finish proposal.
- Write code that determine the point at which it is inefficient to create more threads dividing the data set.

Week 2:

- Write the code to divide up the data and find the maximum, minimum, etc.

Week 3:

- Write the code to sort the data set.
- Compile the code as a library and write a test program.

## Rubric

The grading rubric is shown on the next page.

Part	Unsatisfactory	Satisfactory	Beyond Satisfactory
Determine efficient division of data for threads	Program creates threads and divides up data to an arbitrary block size. [0 – 25 points]	Program determines at what point it would be inefficient to make the block size smaller. [26 – 50 points]	N/A
Determine statistics about the data set	Program does not calculate statistics. [0 – 25 points]	Program calculates the maximum/minimum of the data set and predecessor/successor of an element. [26 – 50 points]	Extra Credit: Program calculates mean, median, and mode averages and standard deviation. [51 – 75 points]
Sort the data set	Program does not sort the data [0 – 25 points]	Program sorts the data in $O(n^2)$ time. [26 – 50 points]	Extra Credit: Program sorts the data in $O(n\log(n))$ time. [51 – 75 points]
Compile as a library	Program does not compile as a usable library. [0 – 25 points]	Program compiles as a library. [26 – 50 points]	N/A