SIT315 – Programming Paradigms

TaskM2.T2C: Complex Threading

Greg McIntyre

218356779

1. Implement a sequential version of the program and use it as your baseline to calculate execution time

https://github.com/gregorymcintyre/ProgrammingParadigms/tree/master/M2.T2C%20-%20Complex%20Threading

qSort.cpp, qSort.h

Quicksort is a common sorting protocol and I implemented a quicksort that meet my needs, there was no reason to reinvent the wheel for a base line calculation so, I choose a simpler more efficient path.

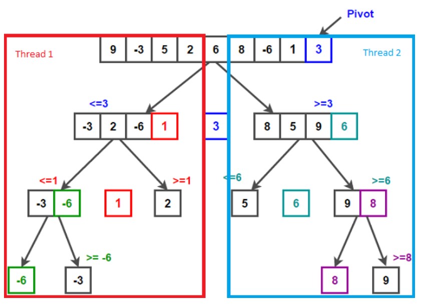
2. Implement parallel version of the program using your choice C / C++ multithreading library - OpenMP or pthreading

https://github.com/gregorymcintyre/ProgrammingParadigms/blob/master/M2.T2C%20-%20Complex%20Threading/ComplexThreading.cpp

I initially attempted to implement a multithread solution using OpenMP (#pragma omp parallel sections, still in code but commented out) but had compiler error and was not able to make it work, so, I implemented a pthread solution.

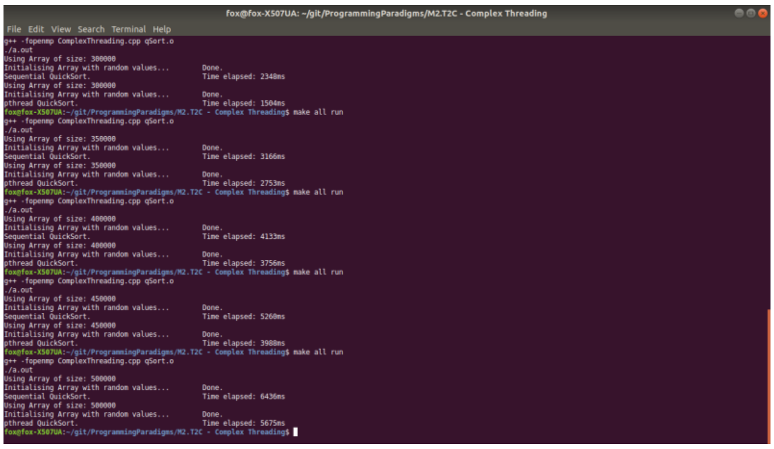
3. Write a document reflecting on the performance of both programs and your analysis of the decomposition you developed.

This was challenging and rewarding task, after reflecting on how the recursion would eventually cause problems with multithreaded tasks I decided to implement a two thread system after the initial pivot. In the below image I have attempted to show how I divided the array to implement a two-thread system. As the two threads do not overlap, it is safe to do so.

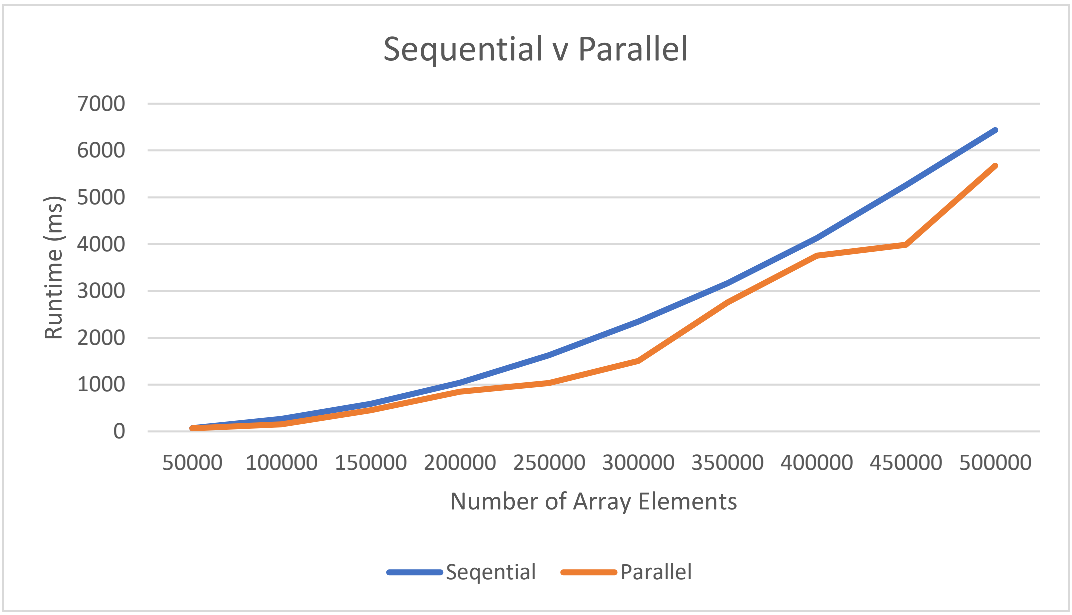


https://www.techiedelight.com/wp-content/uploads/Quicksort.png (Modified)

The program runs well but has an inconsistent improvement on the sequential system.



```
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 300000
Initialising Array with random values...       Done.
Sequential QuickSort.                          Time elapsed: 2348ms
Using Array of size: 300000
Initialising Array with random values...       Done.
pthread QuickSort.                             Time elapsed: 1504ms
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 350000
Initialising Array with random values...       Done.
Sequential QuickSort.                          Time elapsed: 3166ms
Using Array of size: 350000
Initialising Array with random values...       Done.
pthread QuickSort.                             Time elapsed: 2753ms
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 400000
Initialising Array with random values...       Done.
Sequential QuickSort.                          Time elapsed: 4133ms
Using Array of size: 400000
Initialising Array with random values...       Done.
pthread QuickSort.                             Time elapsed: 3756ms
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 450000
Initialising Array with random values...       Done.
Sequential QuickSort.                          Time elapsed: 5260ms
Using Array of size: 450000
Initialising Array with random values...       Done.
pthread QuickSort.                             Time elapsed: 3988ms
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 500000
Initialising Array with random values...       Done.
Sequential QuickSort.                          Time elapsed: 6436ms
Using Array of size: 500000
Initialising Array with random values...       Done.
pthread QuickSort.                             Time elapsed: 5675ms
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ 
```



4. Submit your code and documentation on OnTrack

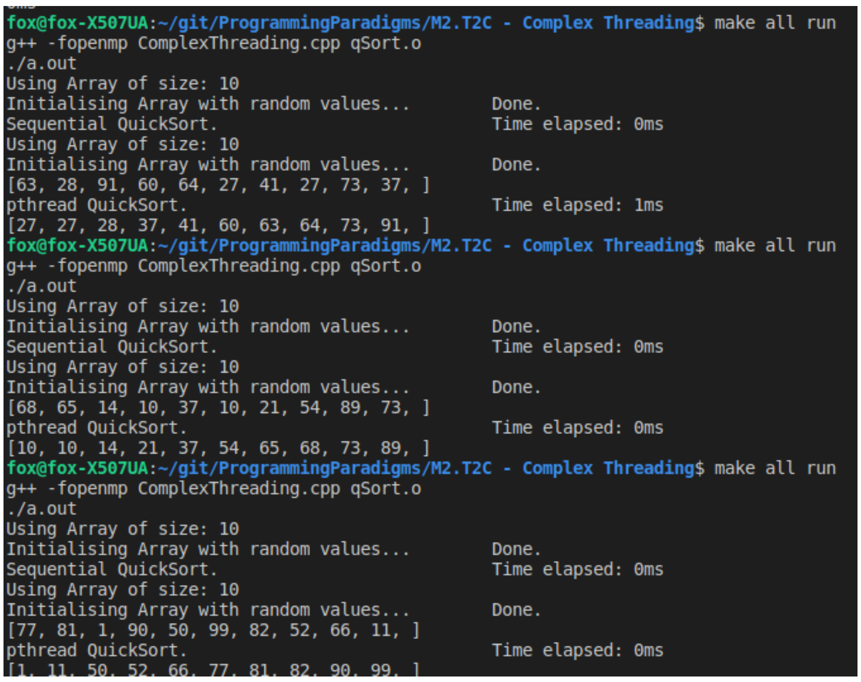
Submitted without qSort.h working git with makefile can be retrieved from git at:

https://github.com/gregorymcintyre/ProgrammingParadigms

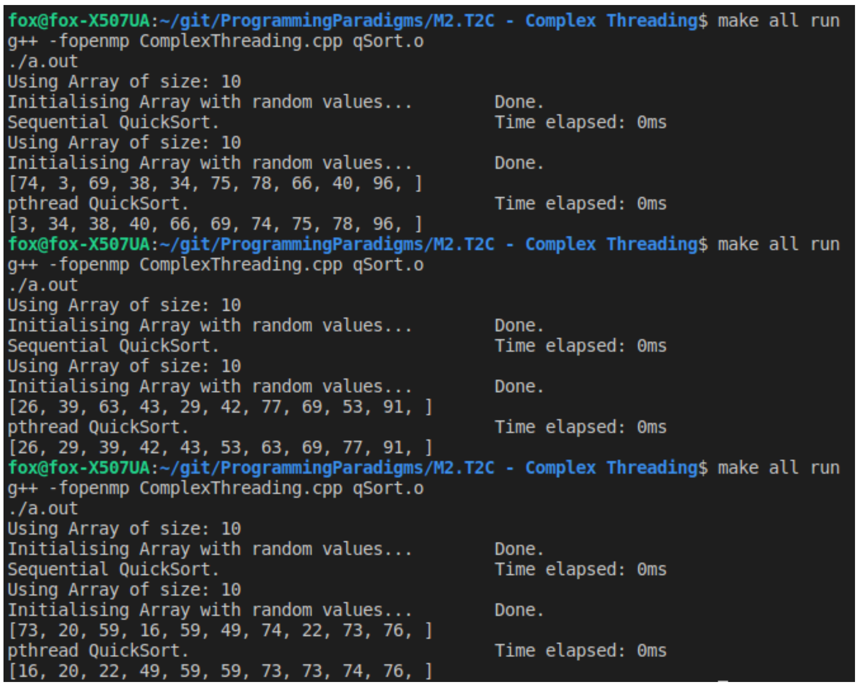
Appendix A: Outputs

“The way your code is parallelised, you will not be getting correct sorting results. You will end up having sorted subdivisions of the array. You divide the array and then sort each subdivision but there is no guarantee that the combined array is sorted, just sorted subdivisions.” MD

“I will double check but I designed it to use the pivot at the division so it will work because the pivot will have smaller to the left and larger to the right. This doesn't make the sides equal as quick sort pivot is often not central but it does multi thread the data.” – GM



```
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 10
Initialising Array with random values...        Done.
Sequential QuickSort.                           Time elapsed: 0ms
Using Array of size: 10
Initialising Array with random values...        Done.
[63, 28, 91, 60, 64, 27, 41, 27, 73, 37, ]
pthread QuickSort.                              Time elapsed: 1ms
[27, 27, 28, 37, 41, 60, 63, 64, 73, 91, ]
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 10
Initialising Array with random values...        Done.
Sequential QuickSort.                           Time elapsed: 0ms
Using Array of size: 10
Initialising Array with random values...        Done.
[68, 65, 14, 10, 37, 10, 21, 54, 89, 73, ]
pthread QuickSort.                              Time elapsed: 0ms
[10, 10, 14, 21, 37, 54, 65, 68, 73, 89, ]
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 10
Initialising Array with random values...        Done.
Sequential QuickSort.                           Time elapsed: 0ms
Using Array of size: 10
Initialising Array with random values...        Done.
[77, 81, 1, 90, 50, 99, 82, 52, 66, 11, ]
pthread QuickSort.                              Time elapsed: 0ms
[1, 11, 50, 52, 66, 77, 81, 82, 90, 99, ]
```



```
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 10
Initialising Array with random values...        Done.
Sequential QuickSort.                           Time elapsed: 0ms
Using Array of size: 10
Initialising Array with random values...        Done.
[74, 3, 69, 38, 34, 75, 78, 66, 40, 96, ]
pthread QuickSort.                              Time elapsed: 0ms
[3, 34, 38, 40, 66, 69, 74, 75, 78, 96, ]
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 10
Initialising Array with random values...        Done.
Sequential QuickSort.                           Time elapsed: 0ms
Using Array of size: 10
Initialising Array with random values...        Done.
[26, 39, 63, 43, 29, 42, 77, 69, 53, 91, ]
pthread QuickSort.                              Time elapsed: 0ms
[26, 29, 39, 42, 43, 53, 63, 69, 77, 91, ]
fox@fox-X507UA:~/git/ProgrammingParadigms/M2.T2C - Complex Threading$ make all run
g++ -fopenmp ComplexThreading.cpp qSort.o
./a.out
Using Array of size: 10
Initialising Array with random values...        Done.
Sequential QuickSort.                           Time elapsed: 0ms
Using Array of size: 10
Initialising Array with random values...        Done.
[73, 20, 59, 16, 59, 49, 74, 22, 73, 76, ]
pthread QuickSort.                              Time elapsed: 0ms
[16, 20, 22, 49, 59, 59, 73, 73, 74, 76, ]
```