Major Project Report

CSCE 735

Sidharth Edupuganti

UIN: 628004560

Compile Arguments:

This C++ program was tested on the Grace machine. The following command is needed to compile the code:

module load intel

icc -qopenmp main.cpp -o main.exe

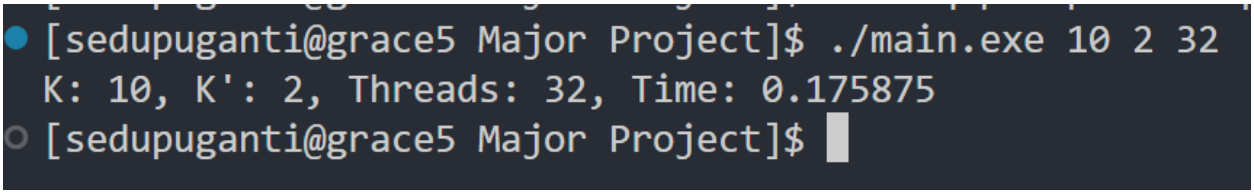
The following command is needed to run the compiled program:

./main.exe <k> <k'> <numOfThreads>

Example:

./main.exe 10 2 32

The output of the program should look something like this:



Batch File Output:

Changing Matrix (2^k) Size:

```
K: 1, K': 1, Threads: 32, Time: 0.0168614
K: 2, K': 1, Threads: 32, Time: 0.00675145
K: 3, K': 1, Threads: 32, Time: 0.0154806
K: 4, K': 1, Threads: 32, Time: 0.00737282
K: 5, K': 1, Threads: 32, Time: 0.00838616
K: 6, K': 1, Threads: 32, Time: 0.00840719
K: 7, K': 1, Threads: 32, Time: 0.0121753
K: 8, K': 1, Threads: 32, Time: 0.0272888
K: 9, K': 1, Threads: 32, Time: 0.0443537
K: 10, K': 1, Threads: 32, Time: 0.251509
```

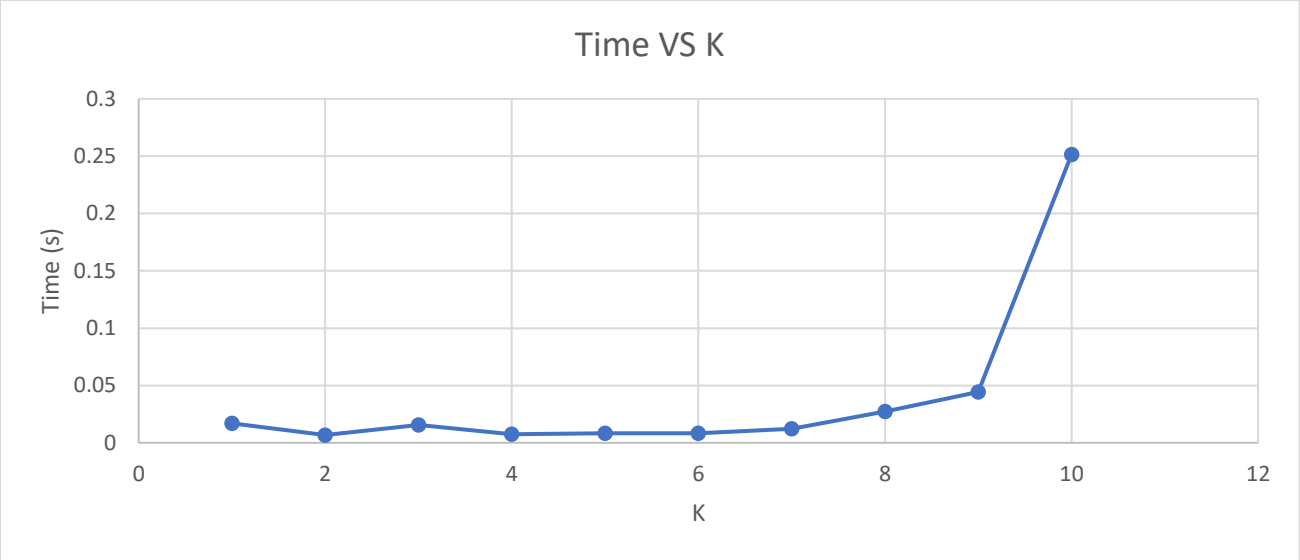
Changing Number of Threads:

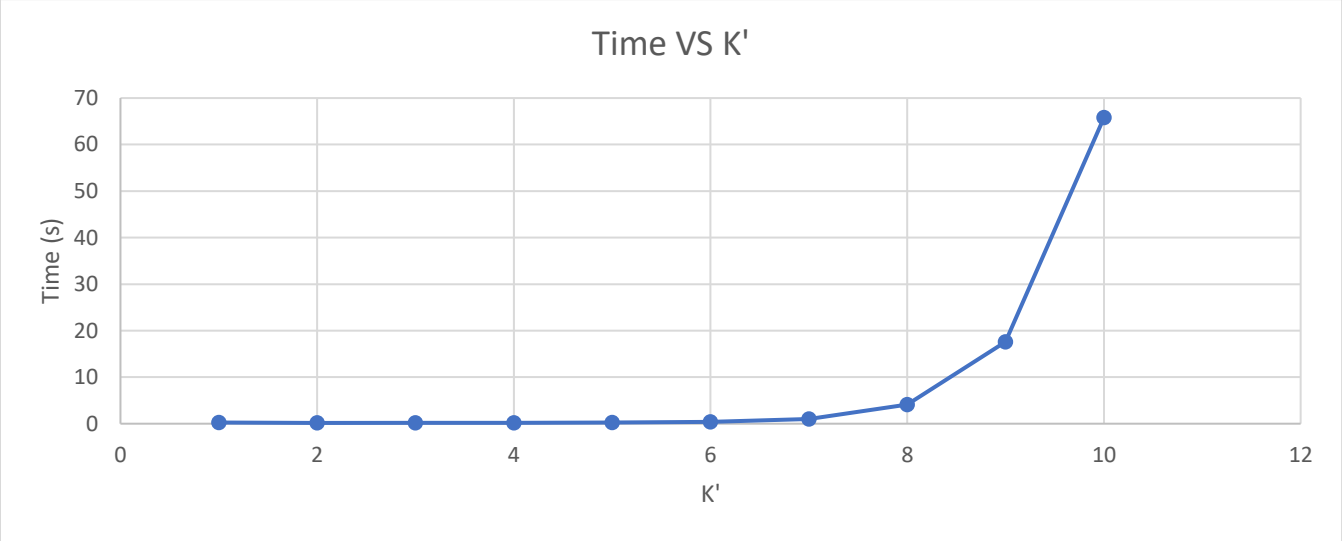
```
K: 10, K': 1, Threads: 1, Time: 1.03101
K: 10, K': 1, Threads: 2, Time: 0.763843
K: 10, K': 1, Threads: 4, Time: 0.418352
K: 10, K': 1, Threads: 8, Time: 0.250516
K: 10, K': 1, Threads: 16, Time: 0.245532
K: 10, K': 1, Threads: 32, Time: 0.257624
K: 10, K': 1, Threads: 64, Time: 0.817865
K: 10, K': 1, Threads: 128, Time: 0.794462
K: 10, K': 1, Threads: 256, Time: 1.6032
K: 10, K': 1, Threads: 512, Time: 1.96393
```

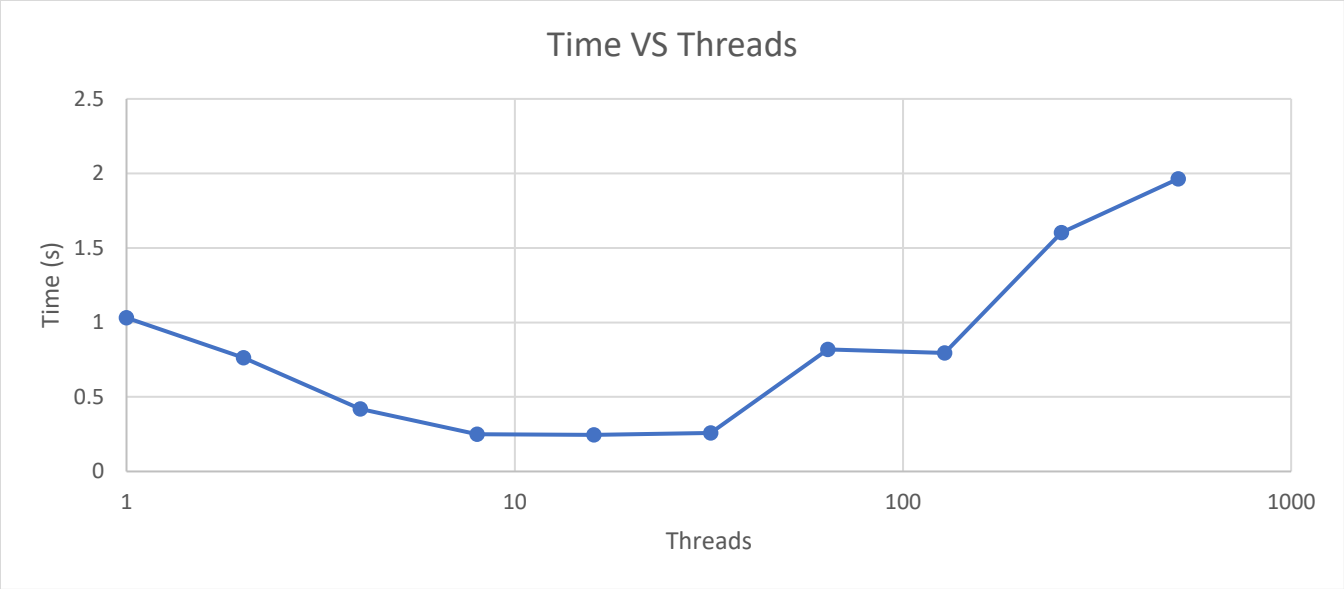
Changing Terminal Matrix (2^k') Size:

```
K: 10, K': 1, Threads: 32, Time: 0.259545
K: 10, K': 2, Threads: 32, Time: 0.159605
K: 10, K': 3, Threads: 32, Time: 0.172287
K: 10, K': 4, Threads: 32, Time: 0.193259
K: 10, K': 5, Threads: 32, Time: 0.240165
K: 10, K': 6, Threads: 32, Time: 0.363289
K: 10, K': 7, Threads: 32, Time: 1.00082
K: 10, K': 8, Threads: 32, Time: 4.13121
K: 10, K': 9, Threads: 32, Time: 17.5742
K: 10, K': 10, Threads: 32, Time: 65.7843
```

Graphs:


Time VS K
Time (s)
0.3
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
12
K


Time VS K'
Time (s)
70
60
50
40
30
20
10
0
0
2
4
6
8
10
12
K'



Discussion:

Based on the graphs, there are a couple things to take away from here. It seems that if K, or K' increase, the time it takes for Strassen to compute the matrix also increases over a period. Both graphs are similar, however K utilizes much less time than compared to K'. The time range for K consisted of being less than 0.5 seconds, while K' time range was between 0 and 70 seconds. This makes sense because K' is the number of levels to reach before the terminal matrix. That means that as K' increases, then the number of recursions will increase, which will cause the computation to take more time to complete. If we look at the Time VS Threads graph, we can see that the K and K' values were kept the same, with the number of threads increasing on each trial run. It seems that the number of threads with the lowest time is around 16 – 32 threads, hence 32 threads were used as a constant for the other two tests (changing K, and changing K'). As the threads increased, the return time also increased due to overhead, which will take away from performance.