# CS33 Homework Assignment 7

*Due 11:59pm Friday, October 25, 2019*

1. A lower bound on the time required for matrix multiplication can be obtained by considering the constraint that the entire multiplier and multiplicand matrices must be transferred from memory to the processor and the resulting product matrix must be transferred from the processor to memory.
   a. Given a maximum memory bandwidth of 10.6 GB/sec for SunLab computers (Intel Core I5-4690 processors with DDR3-1333 memory), what is the lower bound on the time required to multiply two 2048x2048 matrices of doubles? (Assume "mega" means $2^{20}$ and "giga" means $2^{30}$).

   b. This bound is not achievable unless everything fits in cache (and, even then, it wouldn't be easy). Let's assume our processors have minimal caches, and thus everything must be loaded from or stored to memory on most accesses. Assuming the elements of two of the matrices (multiplicand, multiplier, and product) must be accessed $n^3$ times (better algorithms exist, but are, in general, not practical), how much time is required, just for memory access, to multiply two 2048x2048 matrices (i.e., n = 2048)?

   c. The kji algorithm of slide XVII-23 takes 146 seconds on SunLab machines for 2048x2048 matrices of doubles (this isn't the best algorithm we looked at). Explain why so much additional time is required than mentioned in your answer to 2b. (While it does take time to do the necessary multiplications and additions, you may assume this is a rather small part of the difference between 146 seconds and your answer to 2b.) We are not asking you to account for every second of the difference, but to give a qualitative answer explaining why this difference exists. Note that the size of a cache line on SunLab processors is 64 bytes (8 doubles).

2. Consider the following program:

```
int main() {
   for (int i=0; i<=4; i++)
      fork();

   return 0;
}
```

   How many processes are created (including the process that calls *main*)? Do all processes terminate?