# CS575: Programming Assignment 2
## Due at 11:59:59PM, October 19

1. [30%] Implement Strassen's matrix multiplication algorithm. Your program should take an input variable *n* (=$2^k$ where k is a positive integer, $1 \le n \le 1{,}024$) in the Linux command line and generate two n*n random integer matrices, A and B. To avoid the integer overflow, please generate the maximum random integer as $\lfloor \sqrt{maximum\_integer/n} \rfloor$ for each input variable n. Compute A*B using Strassen's algorithm and compare the result to the result produced by the standard matrix multiplication algorithm with $O(n^3)$ time complexity. Print the results, if correct. (If incorrect results are produced, no credit will be given. Your program should work for any matrices. If it works for specific matrices but doesn't work for other matrices, no credit will be given.) Finally, save your source code in a file and name the file as *yourlastname_ yourfirstname_pa2_strassen.cpp*.

2. [30%] Implement Large Integer Multiplication algorithm (Slide 28 of lecture ch5-divide-conquer lecture note). Modify your algorithm so that it divides each n-digit integer into three smaller integers of n/3 digits. Your programs should take a input variable n(=6k where k is a positive integer) in the Linux command line and generate two n-digit random integers (the most significant digit is between 1 and 9, not 0), A and B. Compute A*B using original algorithm discussed in class and the algorithm you modified. Please make sure that you get the same results for the two algorithms. Print the results, if correct. No credit will be given if the algorithm is incorrectly implemented, the time complexity of your program is higher than $O(n^2)$, or your program only works for specific k values. Save your source code in a file and name the file as *yourlastname_yourfirstname _pa2_lim.cpp*.

3. [30%] Implement heap sort and radix sort algorithms. Save your source code in a file and name the file as *yourlastname_yourfirstname _pa2_heap_radix.cpp*. More specifically, your program should do the following (Similar to the first programming assignment):
   a) Let a user select the two *sorting algorithm* to use and specify the number of input data *n* where $1 \le n \le 1{,}000$. Return an error message, if the selected sorting method is not supported by your program or $n < 1$ or $n > 1{,}000$.
   b) Randomly generate an array of n positive integers and print them.
      Note: For radix sort, ensure that the value of each randomly generated data item is an integer that ranges between 0 and 999 (treat each digit as a "digit"). For the other sorting algorithm, data values are arbitrary integers returned by the pseudo random number generator.
   c) Sort the randomly generated data in non-descending order using the selected sorting algorithm. Print the final result.
   d) If $n \le 20$, create random numbers that range between 0 and 15 and do simple visualization of the sorting process as follows. (Do not do this if $n > 20$.) For a number *k* that is randomly generated in Step 2, print "*" *k* times and move to a new line. For example, if the data to sort are (7, 3, 5), your program needs to display the following

before starting to sort them:
*******
***
*****

After executing one step of the selected sorting algorithm, print the array to show the current state. For instance, if the array becomes (3, 7, 5) after executing one step of the selected sorting algorithm, your program needs to print the following:
***
*******
*****

4. [10%] 10% of the grade will be based on *good coding style* and *meaningful* comments.

All programming must be done using **C or C++ in Linux** where your code will be tested. Create a tar file that includes (1) three source code files and (2) a readme file that clearly describes how to compile and run your code. Submit only the tar file through the Blackboard. The name of the tar file should be yourlastname_yourfirstname_pa2.tar (Do not use special characters like #, @, or &, since they have caused Blackboard problems in the past.) Suppose that your assignment files are under the directory of /your_userid/yourlastname_yourfirstname_pa2/ and you are under that directory right now. To create a tar file under /your_userid/ directory, do the following in Linux command line:

```
>cd ..
>tar cvf  your_lastname_ yourfirstname_pa2.tar  yourlastname_yourfirstname_pa2
```

To view the content of the created tar file, do the following in Linux command line:

```
>tar tvf  your_lastname_ yourfirstname_pa2.tar
```

*Finally, read the following policies carefully:*

- *All work must represent your own, individual effort. If you show your code or any other part of your work to somebody else or copy or adapt somebody else's work, you will get zero. To detect software plagiarism, your programs will be checked using Moss (http://theory.stanford.edu/~aiken/moss/).*
- *Your code will be compiled and executed. If your code does not compile or produce any runtime errors such as a segmentation fault or bus error, you will get zero.*
- *The instructor and TA will not read or debug your code. The instructor and TA will not take a look at an emailed code either. If you need general directions, show your code to the TA during his office hours. The TA will not do programming or debugging for you though. He will only help you understand algorithms to be implemented and answer basic questions related to implementation.*