CS575 Design and Analysis of Algorithms

Fall 2023

***Programming Assignment* 2**

Assigned: September 27, 2023
Due: Midnight Tuesday, October 24, 2023

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 *<lastname>_<firstinitial>_pa2_strassen.cpp or <lastname>_<firstinitial>_pa2_strassen.java.*
   Your program should be invoked as: *<lastname>_<firstinitial>_pa2_strassen n*
   Your example output should be (if n = 4):

```
A =
 36   89   57   24
  6   41   49   63
 13  122   98   58
106    5  102   83

B =
91   84  123   97
97   21   44   33
36   16   75   65
88   64   29   90

The standard matrix multiplication A*B =
16073     7341    13315    12294
11831     6181     8044    10790
21649     8934    15999    16877
21107    15953    23315    24547

The Strassen's multiplication A*B =
16073     7341    13315    12294
11831     6181     8044    10790
21649     8934    15999    16877
21107    15953    23315    24547
```

2. [60%] 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 an 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 *<lastname>_<firstinitial> _pa2_lim.cpp or <lastname>_<firstinitial> _pa2_lim.java*.

   Your program should be invoked as: *<lastname>_<firstinitial>_pa2_lim n*
   Your example output should be (if n = 6):
   A = 103456
   B = 200007

   The large integer multiplication from the division of two smaller integers is
   A*B = 20691924192

   The large integer multiplication from the division of three smaller integers is
   A*B = 20691924192

3. [10%] 10% of the grade will be based on *good coding style* and *meaningful* comments.

All programming must be done using **C or C++ or java in Linux** (remote.cs.binghamton.edu) where your code will be tested. Create a .tar.gz 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.gz file through the Brightspace. The name of the .tar.gz file should be <lastname>_<firstinitial>_pa2.tar.gz (Do not use special characters like #, @, or &, since they have caused Brightspace problems in the past.) Suppose that your assignment files are under the directory of /your_userid/<lastname>_<firstinitial>_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  <lastname>_<firstinitial>_pa2.tar  <lastname>_<firstinitial>_pa2
>gzip <lastname>_<firstinitial>_pa2.tar
```

To view the content of the created tar file, do the following in Linux command line:
>tar tvf <lastname>_<firstinitial>_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 TAs will not read or debug your code. The instructor and TAs will not take a look at an emailed code either. If you need general directions, show your code to the TAs during their office hours. The TAs will not do programming or debugging for you though. They will only help you understand algorithms to be implemented and answer basic questions related to implementation.*