Exercise 1 – Strassen's Algorithm

COMP 6651 – Winter 2018

Dr. Tiberiu Popa

Introduction

This exercise will help you acquire practical experience with divide and conquer and recursion. You are asked to implement Strassen's method for multiplying two square matrices. Strassen's method is presented in detail in the course textbook in section 4.2, page 79 in the third edition. It is a very common algorithm and you can find a lot of resources about the algorithm on the web. (https://en.wikipedia.org/wiki/Strassen_algorithm).

Specifications

The input is specified in a file whose name is the first argument of the program. The first line contains an integer M specifying how many datasets are in the file. The reminder of the file encodes the datasets. Each dataset starts with an integer N that is a power of 2 specifying the size of the 2 input matrices (i.e. N by N). The following 2 lines each contain the entries of the two input matrices (N>0). Each matrix is encoded in one line as a space separated sequence of N^2floating point numbers, enumerate row by row.

Here is an example:

```
3
2
1 2 3 4
9 8 7 6
4
1 0 2 3 7 8 -1 9 4 2 0 0 1 -1 2 5
2 8 10 19 21 0 0 -1 -5 2 8 7 7 0 1 2
1
7
8
```

The above example corresponds to the following test cases (in order):

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} * \begin{pmatrix} 9 & 8 \\ 7 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 2 & 3 \\ 7 & 8 & -1 & 9 \\ 4 & 2 & 0 & 0 \\ 1 & -1 & 2 & 5 \end{pmatrix} * \begin{pmatrix} 2 & 8 & 10 & 19 \\ 21 & 0 & 0 & -1 \\ -5 & 2 & 8 & 7 \\ 7 & 0 & 1 & 2 \end{pmatrix}$$

$$7 * 8$$

The output is a file called whose name is the second argument of the program.: Each line encodes the matrix result for each test case in the order in which they were read form the input file.

For example, the output corresponding to the input above is as follows:

```
23 20 55 48
13 12 29 39 250 54 71 136 50 32 40 74 6 12 31 44
56
```

What is given

You are given a simple matrix class *SMatrix* in the file *simple_matrix.hpp.* The *SMatrix* class only implements a R/W access *operator(int row, int column).* You should not modify this file. You have to include this file to your *main.cpp*. Do not modify this file.

Submission

You have to implement your program in plain C/C++ in a file called *main.cpp* that has no dependency other than the standard C/C++ libraries available in a standard Linux system such as the one in the graduate labs and the *simple_matrix.hpp* file given. The code should compile using the command *g++ main.cpp* in anh machine in the graduate labs.

You need to submit the file *main.cpp* **ONLY** using the EAS system. If you submit anything else you will receive a grade of 0. Submit the exercise under "Assignment 1". The due-date is **February 9th, 5pm.** The deadline is soft in that we accept late submissions at a penalty of 20% (of the total marks) for every 24-hour period. For instance, if you submit at 5:01pm on February 9th, you can receive maximum 80% of the maximum grade. If you submit at 5:01pm February 14th , you automatically receive a grade of 0.

This exercise is individual.

Originality and Plagiarism

Strassen's algorithm is a well-known algorithm, whose code undoubtedly can be found on the internet. This is a reminder to everyone that everything you submit must be your original work and you are expected to be able to explain the code in detail if such a request is made by the teaching assistants or the instructor.

Evaluation and Testing

The code is evaluated automatically, but we do check if you implemented the right algorithm and if the code is original. You will receive 0 if the code does not compile. We run 3 test cases, you receive 1/10 marks if your code compiles but it does not return the correct results on any of the test cases, 4/10 if it compiles and runs correctly on one of the test cases, etc.

You will receive a code of 0 if you implemented a different algorithm.

You may be asked to explain your code to the teaching assistant or the instructor. This may lead to a decrease in your overall grade for this exercise.