**Exercise 8. Answer Sheet**

Student's Name: Yuta Nemoto Student's ID: s1240234

***Problem 1.***  Write pseudo-code for the Strassen's algorithm.

STRASSEN(A, B)

// Input: A, B – n x n matrix

// Output: C – n x n matrix

n = |A.rows|

C = new (n×n) matrix

if n == 1

C11 = A11 \* B11

else

/\* Calculate the sum matrices \*/

S1 = B12 – B22

S2 = A11 + A12

S3 = A21 + A22

S4 = B21 – B11

S5 = A11 + A22

S6 = B11 + B22

S7 = A12 – A22

S8 = B21 + B22

S9 = A11 – A21

S10 = B11 + B12

/\* Calculate the product matrices \*/

P1 = STRASSEN(A11, S1)

P2 = STRASSEN(S2, B22)

P3 = STRASSEN(S3, B11)

P4 = STRASSEN(A22, S4)

P5 = STRASSEN(S5, S6)

P6 = STRASSEN(S7, S8)

P7 = STRASSEN(S9, S10)

/\* Calculate the final product sub matrices \*/

C11 = P5 + P4 – P2 + P6

C12 = P1 + P2

C21 = P3 + P4

C22 = P1 + P5 – P3 – P7

return C

***Problem 2.*** Use Strassen's algorithm to compute the matrix product:

Show your work below:

Let A = , B = .

To calculate the sum matrices,

S1 = B12 – B22 = 8 – 2 = **6**

S2 = A11 + A12 = 1 + 3 = **4**

S3 = A21 + A22 = 7 + 5 = **12**

S4 = B21 – B11 = 4 – 6 = **-2**

S5 = A11 + A22 = 1 + 5 = **6**

S6 = B11 + B22 = 6 + 2 = **8**

S7 = A12 – A22 = 3 – 5 = **-2**

S8 = B21 + B22 = 4 + 2 = **6**

S9 = A11 – A21 = 1 – 7 = **-6**

S10 = B11 + B12 = 6 + 8 = **14**

To calculate product matrices,

P1 = A11S1 = 1 \* 6 = **6**

P2 = S2B22 = 4 \* 2 = **8**

P3 = S3B11 = 12 \* 6 = **72**

P4 = A22S4 = 5 \* (-2) = **-10**

P5 = S5S6 = 6 \* 8 = **48**

P6 = S7S8 = (-2) \* 6 = **-12**

P7 = S9S10 = (-6) \* 14 = **-84**

To calculate the final product sub matrices,

C11 = P5 + P4 – P2 + P6 = 48 – 10 – 8 – 12 = **18**

C12 = P1 + P2 = 6 + 8 = **14**

C21 = P3 + P4 = 72 – 10 = **62**

C22 = P1 + P5 – P3 – P7 = 6 + 48 – 72 + 84 = **66**

Then, the result of C = A \* B is

C =

***Problem 3.*** Make two programs implementing the Recursive matrix multiplication and the Strassen's algorithm. Upload your code. Generate two random matrices A and B of size n×n, multiply them using your programs and measure the time needed to get the result. Fill the following table:

Time needed to multiply two n×n matrices. (May depend on the programming language, computer, etc.)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | n | | | | | |
| 32 | 64 | 128 | 256 | 512 | 1024 |
| Recursive (sec) | 0.021 | 0.130 | 0.820 | 5.678 | 46.595 | 407.868 |
| Strassen (sec) | 0.033 | 0.182 | 1.086 | 7.022 | 50.042 | 340.350 |

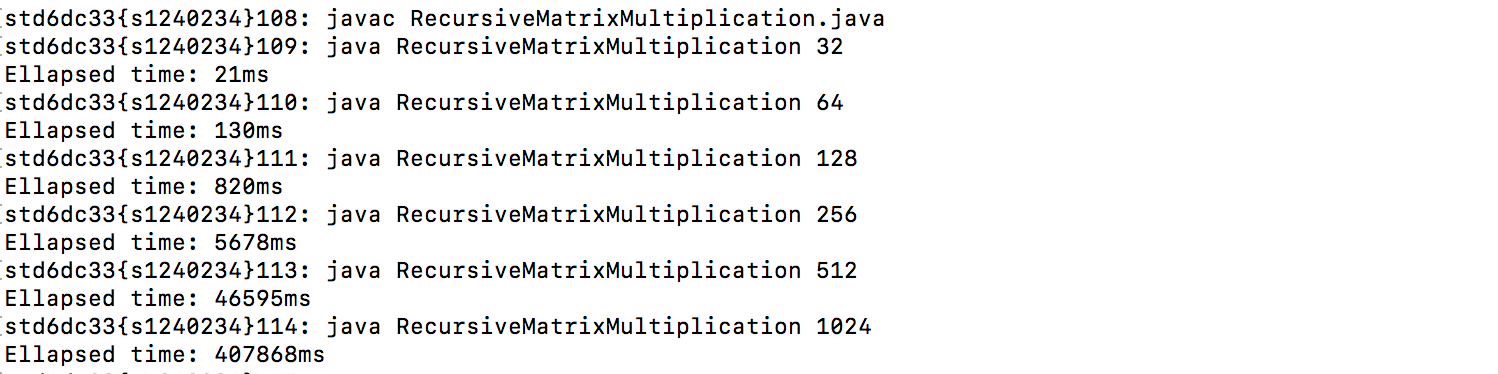
How to compile/run:

1. For the Recursive matrix multiplication code, execute the following:

javac RecursiveMatrixMultiplication.java

java RecursiveMatrixMultiplication [n]

Actual interface is like the screenshot below.

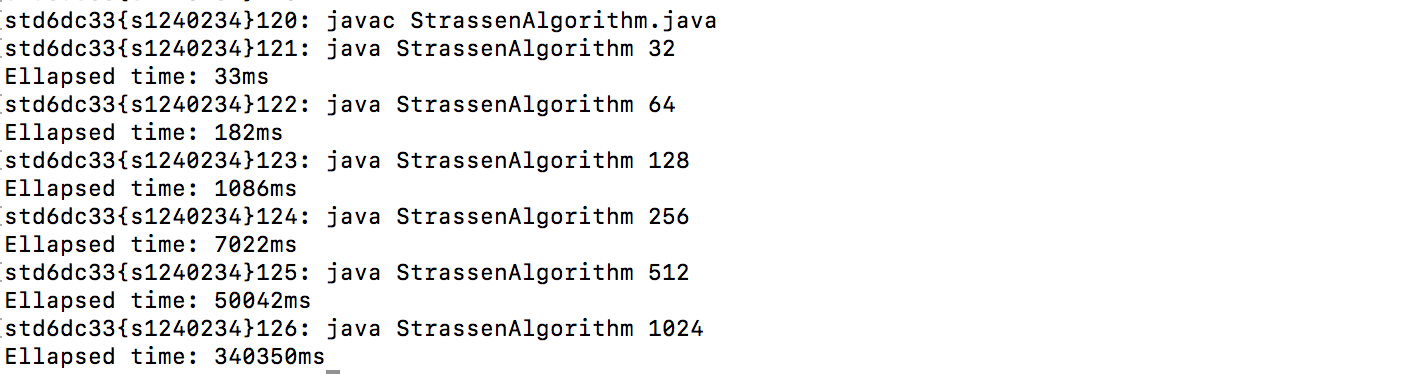


1. For the Strassen’s algorithm code, execute the following:

javac StrassenAlgorithm.java

java StrassenAlgorithm [n]

Actual interface is like the screenshot below.



1. If you want to check the actual result of the matrix calculation, you can check it by adding “-CHECK” to the second argument like below.

