**Exercise 8. Answer Sheet**

Student's Name: Kohei Sato Student's ID: s1240010

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

STRASSEN(A, B)

n = A.rows

let C be a new n\*n matrices

if n == 1

C11 = a11\*b11;

else

partition A, B, P in equations 4.9

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

//Recursive

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)

C11 = P5 + P4 - P2 + P6

C12 = P1 + P2

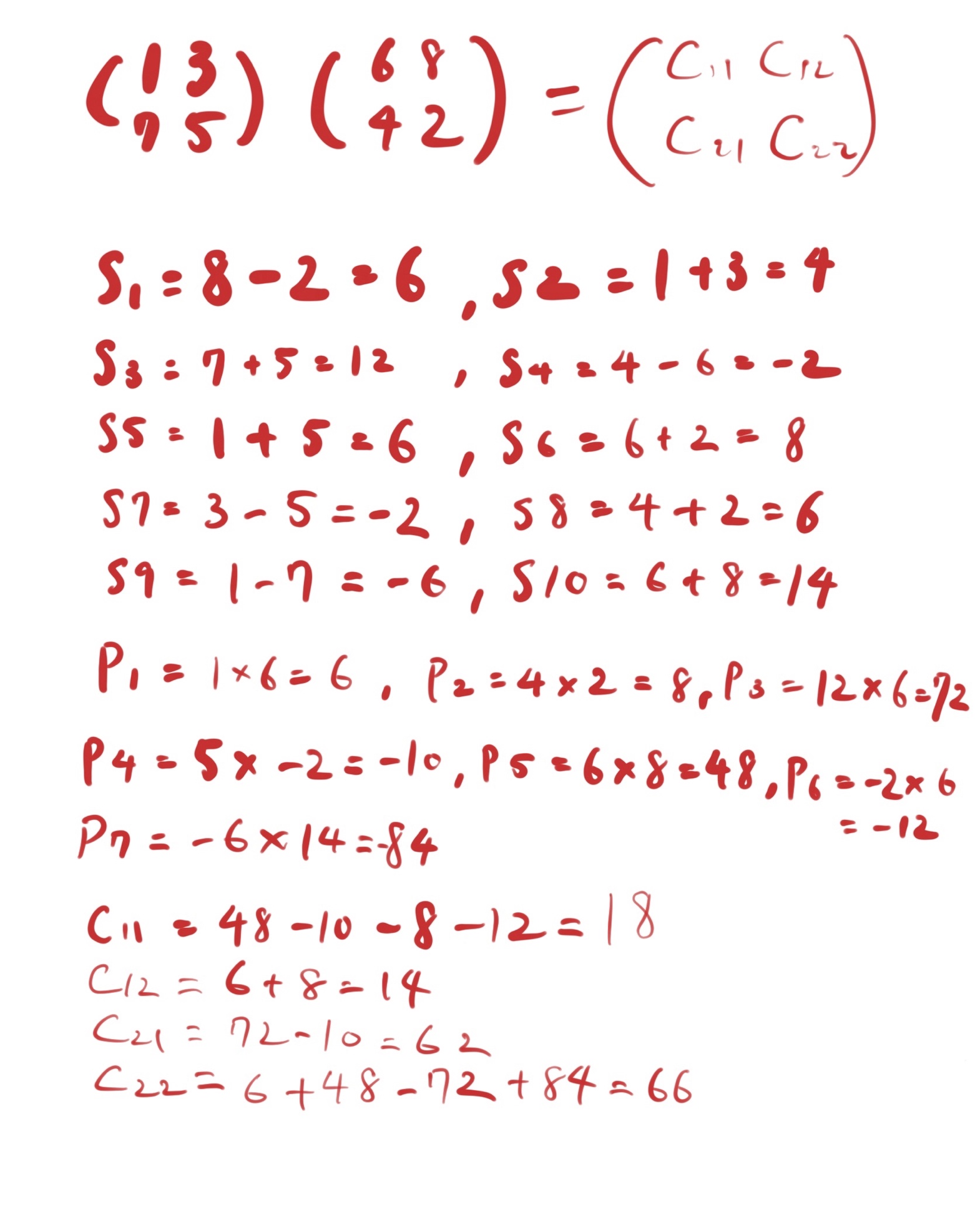
C21 = P3 + P4

C22 = P5 + P1 - P3 - P7

return C

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

Show your work below:



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) | 0.0781898498535 | 0.476845979691 | 4.14728307724 | 37.5658779144 | 371.939133167 | 4710.939133167 |
| Strassen (sec) | 0.0253939628601 | 0.0413360595703 | 0.348987817764 | 2.60540485382 | 21.2877118587 | 176.405590057 |

wlan-napt-001:Ex8 koheisato$ python Recursive.py

Input n:32

0.0781898498535

wlan-napt-001:Ex8 koheisato$ python Strassen.py

Input n:32

0.0253939628601