**ALGORITHM**

**<Strassen algorithm>**

**2017011976 박경리**

**2017012033 배수연**

**2017012197 여채린**

**1. Define the threshold value**

When threshold is set to 2, basic matrix multiplication algorithm is faster than Strassen’s algorithm.

However, when threshold is set to 4, Strassen’s algorithm is faster than basic matrix multiplication algorithm.

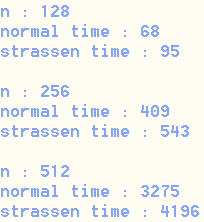
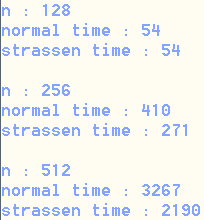
So, value of threshold in our code that the Strassen’s algorithm is faster than the basic algorithm is 4.

(Since the size of the input matrix is a power of two, 3 is not considered.)

Our code is attached to the last page.

<Execution result>

i) threshold = 2 ii) threshold = 4

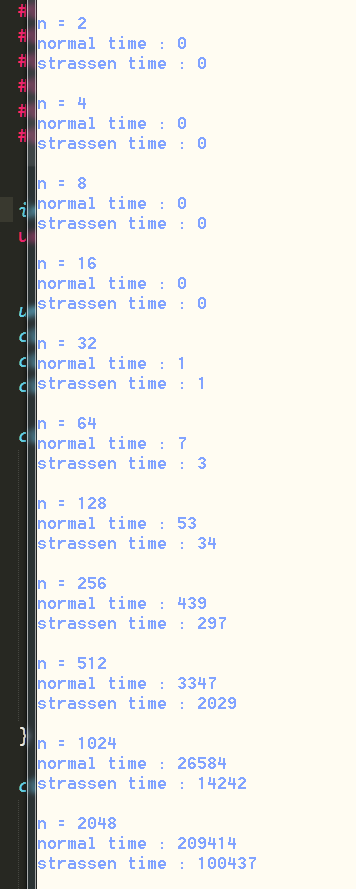
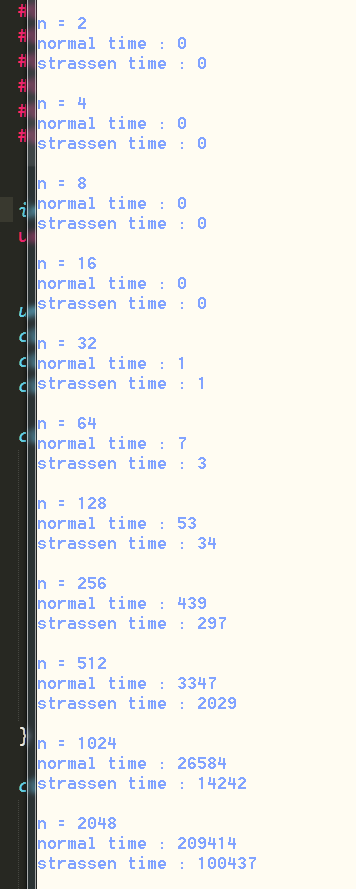
\*n is size of matrix

\*The unit of time is milliseconds

**2. Draw a graph**

**and decide the coefficient in T(n) = C\*n2.81**

<result of Strassen algorithm (threshold = 4)>



\*n is size of matrix

\*The unit of time is milliseconds

In the result, if we calculate C :

(We suppose one operation is executed 100 million times per second.)

n = 256) C = 0.297 / (2562.81/100million) = 5.07

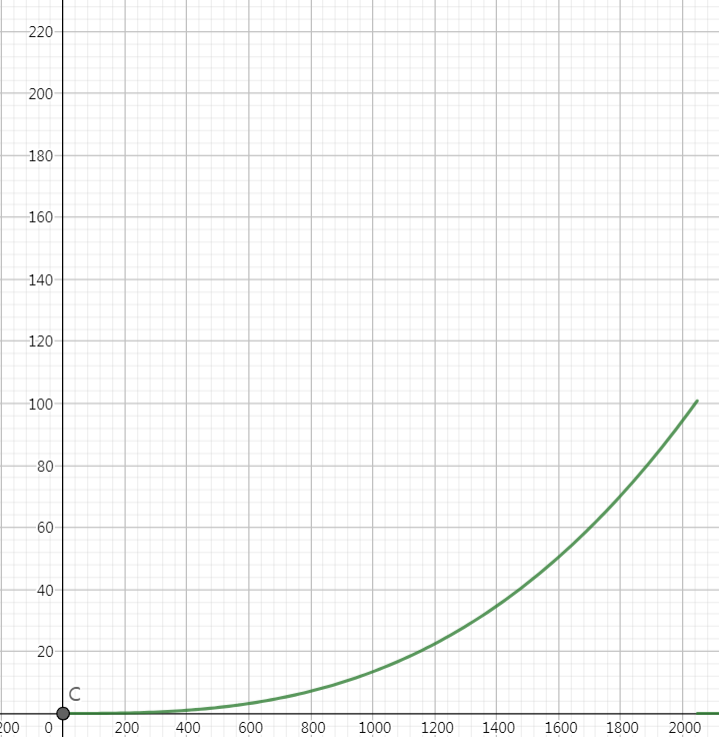
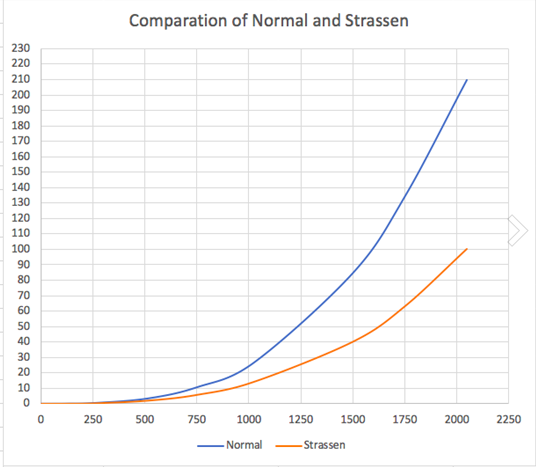
n = 512) C = 2.029 / (5122.81/100million) = 4.95

n = 1024) C = 14.242 / (10242.81/100million) = 4.95

n = 2048) C = 100.437 / (20482.81/100million) = 4.98

So, we estimate C ≒ 5.00±0.05

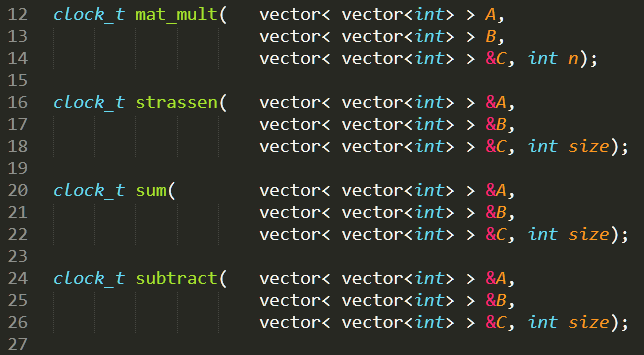
<Graph>



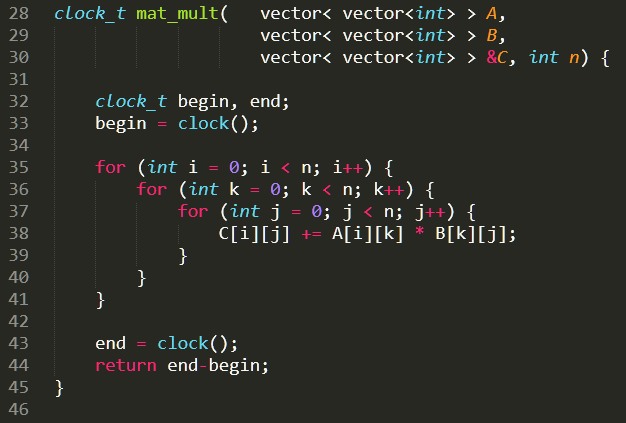
< graph of our result> <graph of T(n) = 5\*n2.81>

# our code

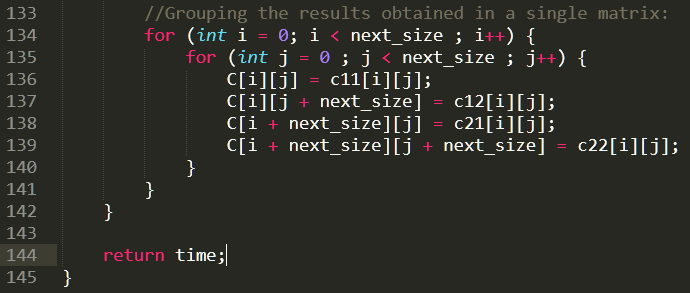
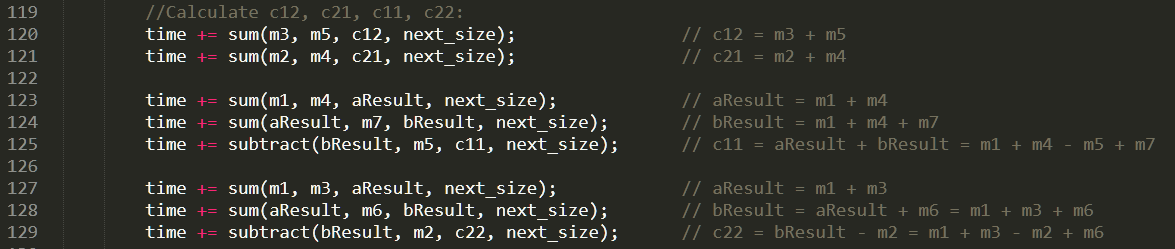
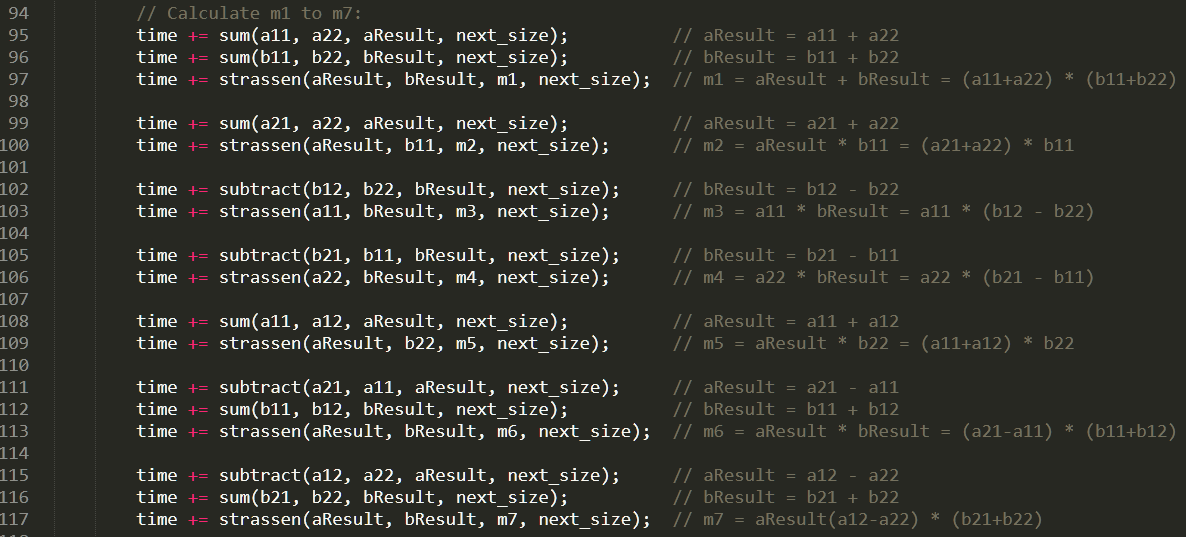
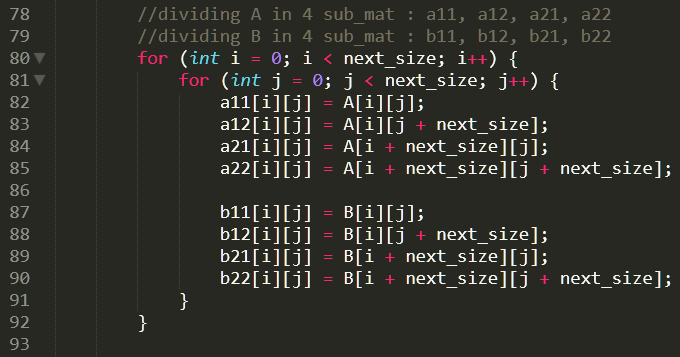
<define>



<basic matrix multiplication algorithm>



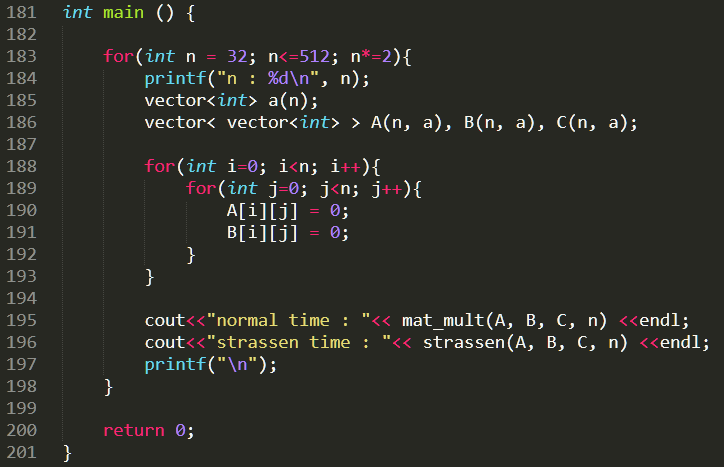
<Strassen’s algorithm>



<matrix sum & subtract>



<main function>

****