***Strassen Algorithm For Matrix Multiplication***

Strassen Algorithm is one of the usage of Divide and Conquer.

***Definition:***

If you have kept contact with Strassen Algorithm before, you may know how to run Matrix Multiplication.

For A = (aij) and B = (bij) are all n \* n Matrix, then for i, j = 1, 2, ..., n, here define cij as the element of Multiplication C = A \* B:

Cij = Sum ( aik \* bkj ) ( k belongs from 1 to n. )

Therefore we need to calculate n^2 number of matrix elements, and each of which is Sum of n values.

***Pseudo Code - Simpler & Slower Version ( No Divide and Conquer Thinking )***

***Pre - Condition:***

* Multiplication of Matrix A and B:
* A, B are all N \* N Matrix.
* Assign C with N \* N Matrix.

*int n = A.rows;*

*For ( int i = 0; i <= n; i ++ )*

*{*

*For ( int j = 0; j <= n; j ++ )*

*{*

*C[ i ] [ j ] = 0;*

*For ( k = 0; k <= n; k ++ )*

*{*

*C [ i ] [ j ] += A [ i ] [ k ] \* B [ k ] [ j ];*

*}*

*}*

*}*

*RETURN C;*

***Cost:***

O( N ^ 3 )

***Supplement:***

You may probably think that any Matrix Multiplication need to spend O ( N ^ 3 ) cost, but actually, we have the Divide and Conquer Algorithm which can help us speed up. It only cost O ( n ^ lg7 ).

***The Simpler Divide and Conquer Version:***

When we try to calculate Matrix Multiplication C = A \* B by using Divide and Conquer, then assume that three Matrix are all n \* n, also n is Power of 2. Here we assume that in each division steps, then n \* n matrix are divided into 4 n/2 \* n/2 Sub - Matrix.

***Example:***

|  |  |
| --- | --- |
| A11 | A12 |
| A21 | A22 |

|  |  |
| --- | --- |
| B11 | B12 |
| B21 | B22 |

A = B =

|  |  |
| --- | --- |
| C11 | C12 |
| C21 | C22 |

C =

Therefore, we can modify the final Expression, which is:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| C11 | C12 | = | A11 | A12 | \* | B11 | B12 |
| C21 | C22 | A21 | A22 | B21 | B22 |

So, it can be converted into four equations, just as below:

C11 = A11 \* B11 + A12 \* B21

C12 = A11 \* B12 + A12 \* B22

C21 = A21 \* B11 + A21 \* B21

C22 = A21 \* B12 + A21 \* B22