# Running time analysis of recursive matrix multiplication via the divide and conquer approach

## Theory

From the algorithm provided in the book page 77, it is evident that there would be a total of 8 recursive calls, hence

$a = 8$

Also, we divide the problem set into half in each recursive call (n/2), hence $b = 2$

Now, according to the masters theorem $T(n) = 8T(n/2) + O(n^2)$
Now, case 1 of master theorem states that:

If $f(n)$ is $O(n^{\log_b a-E})$, then $T(n)$ is $\Theta(n^{\log_b a})$ where $E > 0$

In our case, $\text{Log}_2 8 = 3$

Hence, $T(n) = O(n^3)$

## Findings

We run the program in a loop with iteration size upto 2048. The loop will run for each number that is a power of 2 i.e. 2,4,8,16,32,64, etc upto 2048 and will record the time it will take for the algorithm to execute.
This data is being recorded in the “results.txt” file in the program directory and using this file, we plot the running time of the algorithm for each test case, given below.