Problem 1 Exercise 3.1-1. (page 50, textbook). Let $f(n)$ and $g(n)$ be asymptotically non-negative functions. Based on the definition of the $\Theta$ notation, prove that

$$\max(f(n), g(n)) = \Theta(f(n) + g(n))$$

Hint: look up the definition of $\Theta$, and apply it directly.

Problem 2 Order the following functions in the ascending order of their asymptotic growth.

$$5n^2 + 7\frac{n^2}{\log(n)}, 19 \cdot 2\log n^3, n \log(\log(n)), \sqrt{n} \cdot \log(n), 5n^3 + 4n^2 + 7n, 2^{2n}, 12n^7 + 2^{n\log(n)}$$

Problem 3 Express the asymptotic behavior of each of the functions below using the Big-O notation.

1. $n \log(n) + 37n + 17\log(n) + 2 \cdot \frac{n}{\log(n)}$
2. $n^3 + 6n^5 + 12n^4 + 6n^3 + n^2 + 2n$
3. $n^3 + n^3 \log(n)$
4. $\log \log \log(n) + \log(n) + \frac{n}{\log(n)}$
5. $e^{0.5n} + 35n^7$

Problem 4 Problem 3-3.a (pages 58, textbook).

Hint: It is almost like sorting (i.e., you can compare any two functions based on their asymptotic behavior: the $\Omega$, $\Theta$ and $O$ notation representing the $\leq$, $=$ and $\geq$ comparisons.

Note: Look up the definition of $\log^*$ in the textbook, if you are not sure.

Problem 5

In class, we have studied two selection algorithms: finding the largest and the second largest elements of an array. Using the same intuition as for the $\text{FindSecondMax}()$ algorithm for finding the second largest element of an array (using tournaments), propose an algorithm $\text{findThirdLargest}()$.

1. Describe the algorithm using pseudocode.
2. Analyze the number of comparisons the algorithm will use.

Note: Your algorithm shall use (asymptotically) fewer comparisons than the brute-force algorithm of selecting the largest number in an array three times in a row.

Problem 6 Problem 4-1, textbook (page 85).
Problem 7  Consider the following array:

\[ A = [6, 19, 5, 17, 33, 56, 22, 42, 31, 6, 21, 90, 32, 65, 13] \]

1. Using recursion trees, show the work of the \textbf{MergeSort} algorithm on \( A \) as input.

2. Using recursion trees, show the work of the divide-and-conquer \textbf{FindMedian} algorithm on \( A \) as input.

Problem 8  Illustrate the work of Strassen’s algorithm for matrix multiplication on the following pair of matrices:

\[
A = \begin{pmatrix} 2 & 4 & 6 & 2 \\ 3 & -2 & 1 & -2 \\ 7 & 1 & -1 & 3 \\ 2 & 0 & 5 & -2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & -3 & 4 & 2 \\ -1 & -2 & 4 & 0 \\ 2 & 0 & -6 & 0 \\ 1 & 1 & 2 & -2 \end{pmatrix}
\]

Problem 9  Show that the padding of matrices \( A \) and \( B \) of size 3x3 to matrices of size 4x4 with zeros results in a correct output of both the Divide-and-Conquer and the Strassen’s algorithms when given the padded versions of \( A \) and \( B \) as the input matrices.

\textbf{Hint:} this may be time consuming, but basically, just compute the matrices both algorithms would return when the 4x4 padded matrices are provided to them as input.