MAS 714, Fall 2019
Tutorial 1

Problem 1 Consider the following functions, and describe their relationship in O and Ω notation (logarithms have base 2). For this list them in asymptotically increasing order.

1. $f_1(n) = 7n - 2$
2. $f_2(n) = n^{\sqrt{1/\log n}}$
3. $f_3(n) = 100\sqrt{n} + n^{1/4}/2 - 1$
4. $f_4(n) = 2^{\log^2 n}$
5. $f_5(n) = 2^{2\log n} + n^{1.1}$

Problem 2 Consider two numbers with 2 digits each, i.e., $x = x_1 \cdot B + x_0$ and $y = y_1 \cdot B + y_0$. The base of the number representation is B. It is obvious that xy can be computed using 4 multiplications of pairs of numbers with 1 digit each ($x_1 \cdot y_0$ etc.).
Show that the product xy can also be computed from 3 products of numbers (either x_0, x_1, y_0, y_1 or sums of these).
Use this to find a recursive algorithm for integer multiplication that is faster than $O(n^2)$.
HINT: split n-bit numbers into two $n/2$-bit numbers and apply the above idea to them.

Problem 3 Show that the second smallest of n numbers stored in an array can be found with at most $n + \log n$ comparisons.
HINT: First find the minimum in such a way that there are at most $\log n$ candidates for the second smallest element left. These candidates must be elements that have been compared to the minimum.

Problem 4 The *median* of a sequence of distinct numbers $x_1, \ldots, x_n$ is the element x_i such that $\lfloor n/2 \rfloor$ elements x_j are smaller than x_i and $\lfloor n/2 \rfloor$ elements x_j are larger than x_i (n is assumed to be odd).
Describe a Divide and Conquer algorithm that computes the median of a sequence of numbers stored in an array A in way that is similar to Quicksort. Analyze the worst case and average case/expected running time. The algorithm should be faster than Quicksort (when considering expected time).

Problem 5 Show that Quicksort can be implemented *in place*, i.e., show that there is a partition procedure that splits up the array A (given a pivot element $A[i]$) into the smaller and larger elements only using the array A.