# ADS 2021: Week 4 Exercises

Exercises for Algorithms and Data Structures at ITU. The exercises are from *Algorithms, 4th Edition* by Robert Sedgewick and Kevin Wayne unless otherwise specified. This week also includes exercises from *Algorithms and Data Structures* by Kurt Mehlhorn and Peter Sanders. Color-coding of difficulty level and alterations to the exercises (if any) are made by the teachers of the ADS course at ITU.

**2.1.1 Selection Sort - Green** Show in the style of the example trace with Algorithm 2.1 from the book [SW 2.1], how selection sort sorts the array `E A S Y Q U E S T I O N`.

**2.1.4 Insertion Sort - Green** Show in the style of the example trace with Algorithm 2.2 from the book [SW 2.1], how insertion sort sorts the array `E A S Y Q U E S T I O N`.

**2.2.2 Merge Sort - Green** Show in the style of the example trace with Algorithm 2.4 from the book [SW 2.2], how top-down mergesort sorts the keys `E A S Y Q U E S T I O N`.

**2.1.6 Identical Keys - Green** Which method runs faster for an array with all keys identical, selection sort or insertion sort?

**2.1.7 Reverse Order - Green** Which method runs faster for an array in reverse order, selection sort or insertion sort?

**2.1.8 Random Order - Yellow** Suppose we use insertion sort on a randomly ordered array where items have only one of three values. Is the running time linear, quadratic or something in between?

**2.1 [Mehlhorn-Sanders] Analysis - Yellow** Recall that if there exist constants $c$ and $N_0$ such that for all $n > N_0$, we have $|f(n)| \leq c|g(n)|$, then we write $f(n) = O(g(n))$ (see [SW p. 206]).

**Example:** $n^2 + n = O(n^2)$. *Let* $c = 2$ *and* $N_0 = 1$. *Observe*

$$n^2 + n \leq 2n^2 \quad \Leftrightarrow \quad n \leq n^2 \quad \Leftrightarrow \quad 1 \leq n.$$

*(Side-note: it holds for* $N_0 = 0$ *if we know that* $n$ *is an integer)*

For each of the following statements, decide if it is right or wrong. If it is right, find constants $c$ and $N_0$ to prove your answer.

a) $17 = O(1)$.

b) $2n + \log(n) = O(n)$.

c) $n^2 + 10^6 n = O\left(n^2\right)$.

d) $n \log(n) = O(n)$.

e) $10n + \sqrt{n} = O(n)$.

**2.2.15 Bottom-up Queue Mergesort - Yellow** Explain how to take two queues of sorted items and merge them into a single queue which is still in sorted order.

**2.1.13 Deck Sort - Yellow** Explain how you would put a deck of cards in order by suit (in the order spades, hearts, clubs, diamonds) and by rank within each suit, with the restriction that cards must be laid out face down in a row, and the only allowed operations are to check the values of two cards and to exchange two cards (keeping them face down).

**2.1.14 Dequeue Sort - Red** Explain how you would sort a deck of cards, with the restriction that the only allowed operations are to look at the values of the top two cards, to exchange the top two cards, and to move the top card to the bottom of the deck.

**2.2 [Mehlhorn-Sanders] $O$-notation - Red** Argue for the following statements:

a) If $f(n) = O(g(n))$, then $f(n) + g(n) = O(g(n))$.

b) $O(f(n)) \cdot O(g(n)) = O(f(n) \cdot g(n))$.