Problem 1b:

Let us analyze the pseudocode in (1a). There are two nested loops. The for-loop is a definite loop and therefore we know that it iterates n-times. (n is the size of the array). In worst case there can be n swap operations inside the for-loop. That case is when the array is sorted in descending order. Consider the example:

arr = [5, 4, 3, 2, 1]

$\xrightarrow{\text{after 1 iteration}}$ [4, 3, 2, 1, 5] $\xrightarrow{\text{after 1 iteration}}$ [3, 2, 1, 4, 5] and so on until arr = [1, 2, 3, 4, 5]

In this case the while loop will iterate n times. All other operations take constant time, therefore: $f(n) = cn^2$ $g(n) = n^2$

$T(n) = O(n^2)$ $O(g(n)) = f(n)$ if $\exists c_1, n_0 \in \mathbb{N}$ $0 \leq f(n) \leq c g(n)$

$\Rightarrow 0 \leq cn^2 \leq c_1 n^2 \Rightarrow 0 \leq c \leq c_1$ □

Average case happens when the while loop repeats around $\frac{n}{2}$ times. Then we will have $\frac{n}{2} n$ operations which makes

$f(n) = \frac{n^2}{2} c$

$g(n) = n^2$

$\cancel{0 \leq c_1 g(n) \leq f(n) \Rightarrow 0 \leq c_1 n^2 \leq \frac{n^2}{2} c}$

$\cancel{\Rightarrow 0 \leq c_1 \leq}$

$c_1, c_2 \in \mathbb{N}$

$\cancel{\Theta}$

$c_1 g(n) \leq f(n) \leq c_2 g(n)$

$c_1 n^2 \leq f(n) \leq c_2 n^2$

$c_1 \leq \frac{1}{2}c \leq c_2 \quad \square \quad \Rightarrow \quad T(n) = \Theta(n^2)$

For the best case the array is already sorted, therefore no swap operations which leads us to no more than one repetition of the while-loop.

$f(n) = nc$

$g(n) = n^2 \leftarrow$ guess

$c_1, c \in \mathbb{N}$

$0 \leq c_1 g(n) \leq f(n)$

$\Rightarrow \quad 0 \leq c_1 n^2 \leq nc$

$\Rightarrow \quad 0 \leq c_1 \leq \frac{1}{n}c \quad \rightarrow$ as $n \rightarrow \infty$, $\frac{1}{n}c \rightarrow 0$, therefore

$g(n) \neq n^2$,

let us try $g(n) = n$

$0 \leq c_1 n \leq \frac{n}{2}c$

$0 \leq c_1 \leq \frac{1}{2}c \Rightarrow T(n) = \Omega(n)$