HS22

Claim	true	false
There exists an array of length n for which the runtime of `InsertionSort` is $\Theta(n^{1.5})$.	☐	☐
There exists an array of length n for which the runtime of `MergeSort` is $\Theta(n)$.	☐	☐
There exists an array of length n for which the runtime of `HeapSort` is $\Theta(n^2)$.	☐	☐
Suppose a sequence of n bits (every element is either a zero or one) is given as input. There exists an algorithm with runtime $O(n)$ which sorts any such sequence.	☐	☐

HS21

Sorting algorithms quiz: For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer, 0P for a missing answer. You get at least 0 points in total.

Claim	true	false
The runtime of `MergeSort` on the input $[1, 2, \ldots, n]$ is $\Theta(n)$.	☐	☐
The runtime of `InsertionSort` on the input $[1, 2, \ldots, n]$ is $\Theta(n)$.	☐	☐
The runtime of `InsertionSort` on the input $[n, n-1, \ldots, 1]$ is $\Theta(n \log n)$.	☐	☐

FS21

Let $A[0, \ldots, n-1]$ be an integer array of size n. Consider the following implementation of insertion sort:

Algorithm 1 `InsertionSort`(A)

for $i = 1 \ldots n-1$ **do**
    $B \leftarrow A[i]$
    Find the smallest index $j \in \{0, \ldots, i\}$ such that $A[i] \leq A[j]$.
    Shift the subarray $A[j, \ldots, i-1]$ by one to the right, and move the element B to position j.

Consider the following invariant $INV(i)$: After the ith iteration, $A[0, \ldots, i]$ is sorted.

For each of the following claims, state whether it is true or false. You get 1P for a correct answer, -1P for a wrong answer and 0P for a missing answer. In total, you get at least 0 points.

Claim	true	false
$INV(i)$ holds after the ith iteration of the `for`-loop.	☐	☐
$INV(i)$ can be used to prove the correctness of `InsertionSort`.	☐	☐
$INV(1)$ already holds before the first loop iteration is executed.	☐	☐
At the start of the ith loop iteration, $A[0, \ldots, i-1]$ is sorted. Further, for the smallest index $j \in \{0, \ldots, i\}$ that satisfies $A[i] \leq A[j]$, the following holds: All elements in $A[0, \ldots, j-1]$ are less than $A[i]$ and all elements in $A[j, \ldots, i-1]$ are greater than or equal to $A[i]$. Thus, shifting $A[j, \ldots, i-1]$ by one to the right and moving B to position j yields a sorted subarray $A[0, \ldots, i]$.	☐	☐
After the $(n-1)$th loop iteration $INV(n-1)$ holds and per definition this implies that $A[0, \ldots, n-1]$ is sorted.	☐	☐

HS20

d) *Insertion sort invariant*

Let $A[0,\dots,n-1]$ be an integer array of size n. Consider the following implementation of insertion sort:

Algorithm 2 `InsertionSort`(A)

for $i = 1 \dots n-1$ **do**
  Find the smallest index $j \in \{0,\dots,i\}$ such that $A[i] \le A[j]$.
  Shift the subarray $A[j,\dots,i-1]$ by one to the right, and move the element $A[i]$ to position j.

i) Formulate an invariant $INV(i)$ that holds after the ith iteration of the `for`-loop (the iteration with $i = 1$ is the first iteration).

ii) Use this invariant to prove correctness of the algorithm `InsertionSort`.

1. Show that the invariant holds at the beginning (base case).

2. Let $1 \le i \le n-2$. Show that if $INV(i)$ holds after the ith iteration of the `for`-loop, then $INV(i+1)$ holds after the $(i+1)$st iteration (induction step).

3. Show that if $INV(n-1)$ holds at the end of the algorithm, then the array A is sorted.