Reminder: \( \log n \) denotes the binary logarithm, i.e., \( \log n = \log_2 n \). We also write \( \log^c n \) as an abbreviation for \( (\log n)^c \).

Make sure to put your name on every piece of paper that you hand in!

1 **\( O \)-notation**

1. Let \( f : \mathbb{N} \rightarrow \mathbb{N} \) be a function. Define the set \( O(f(n)) \).

2. Give a formal proof of the statement:

\[
2n^2 + 5n \in O(n^2) .
\]

3. For each of the following statements, indicate whether it is true of false: (no justification needed)

   (a) \( 10n \in O(n \log n) \)
   (b) \( \log^2 n \in O(n^3) \)
   (c) \( \frac{1}{2} \log n \in O(\sqrt{\log n}) \)
   (d) \( n! \in O(\sum_{i=0}^{n} 2^i) \)
   (e) \( 2^{\log \log n} \in O(\log^2 n) \)
   (f) \( f(n) \in \Theta(g(n)) \) implies \( g(n) \in \Omega(f(n)) \)
   (g) \( f(n) \in O(g(n)) \) implies \( 2^{f(n)} \in O(2^{g(n)}) \)

2 **Sorting**

1. What is the runtime of Insertionsort in \( \Theta \)-notation (our aim is to sort the input in increasing order) on the following array of length \( n \): (no justification needed)

\[
A[i] = 1, \text{ if } 0 \leq i \leq \lfloor \frac{n}{2} \rfloor, \text{ and } A[i] = 0 \text{ otherwise}
\]

2. Heapsort interprets an array of length \( n \) as a binary tree. What is the runtime of the \texttt{BUILD-HEAP()} operation that transforms the initial binary tree into a heap? (no justification needed)

3. Suppose that Mergesort is executed on an array of length \( 2^k \), for some integer \( k \). What is the height of the corresponding recursion tree of this execution in \( \Theta \)-notation? (no justification needed)

Continued on next page...
3 Loop Invariant

Consider the following algorithm: (it operates on an array $A$ of length $n$ of positive integers)

**Algorithm 1**

**Require:** $A$ is an array of $n$ positive integers

1: $x \leftarrow A[0]$

2: for $i \leftarrow 1, \ldots, n - 2$ do

3: $x \leftarrow x \cdot A[i]$

4: end for

5: return $x$

Consider the following loop invariant:

At the beginning of iteration $i$ (i.e., after $i$ is updated in Line 2 and before the code in Line 3 is executed) the following property holds:

$$x = \frac{A[0]}{A[i]}.$$  

1. **Initialization:** Argue that at the beginning of the first iteration, i.e. when $i = 1$, the loop invariant holds.

2. **Maintenance:** Suppose that the loop invariant holds at the beginning of iteration $i$. Argue that the loop invariant then also holds at the beginning of iteration $i + 1$.

3. **Termination:** What does the algorithm compute? Argue that this follows from the loop invariant.