Reminder: \( \log n \) denotes the binary logarithm, i.e., \( \log n = \log_2 n \).

1 Proofs by Induction

Prove the following statements by induction:

1. For every integer \( n \geq 0 \), the following holds:

\[
\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.
\]

Solution.

Base case: \( (n = 0) \)

Observe that \( \sum_{i=0}^{0} i^2 = 0 \) and \( \frac{0(0+1)(2\cdot0+1)}{6} = 0 \). The base case thus holds.

Induction Hypothesis: \( \sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \) holds for some value of \( n \).

Induction Step: We need to show that the statement also holds for \( n + 1 \):

\[
\sum_{i=0}^{n+1} i^2 = (n+1)^2 + \sum_{i=0}^{n} i^2 = (n+1)^2 + \frac{n(n+1)(2n+1)}{6} = \frac{(n+1)(6(n+1) + n(2n+1))}{6} = \frac{(n+1)(6n + 6 + 2n^2 + n)}{6} = \frac{(n+1)(n+2)(2n+3)}{6} = \frac{(n+1)(n+2)(2(n+1) + 1)}{6}.
\]

✓

2. For every \( n \geq 1 \), the following holds:

\( 11^n - 6 \) is divisible by 5.

Solution.

Base case: \( (n = 1) \)

Observe that \( 11^1 - 6 = 5 \) which is divisible by 5. The base case thus holds.

Induction Hypothesis: \( 11^n - 6 \) is divisible by 5, for some value of \( n \).
Induction Step: We need to show that the statement also holds for \( n + 1 \):

\[
11^{n+1} - 6 = 11 \cdot 11^n - 6 = 10 \cdot 11^n + 11^n - 6 = 5(2 \cdot 11^n) + 11^n - 6.
\]

Observe that the term \( 5(2 \cdot 11^n) \) is divisible by 5. Furthermore, \( 11^n - 6 \) is divisible by 5 by the induction hypothesis. Since the sum of two numbers that are divisible by 5 is also divisible by 5, the result follows.

\[\checkmark\]

3. Consider the following sequence: \( s_1 = 1, s_2 = 2, s_3 = 3, \) and \( s_n = s_{n-1} + s_{n-2} + s_{n-3} \), for every \( n \geq 4 \). Prove that the following holds:

\[
s_n \leq 2^n.
\]

Solution.

Base cases: We need to verify that the statement holds for \( n \in \{1, 2, 3\} \), since \( s_n \) depends on \( s_{n-1}, s_{n-2}, s_{n-3} \) (in particular, \( s_4 \) depends on \( s_3, s_2, s_1 \)). This is easy to verify: \( s_1 = 1 \leq 2^1, s_2 = 2 \leq 2^2 \) and \( s_3 = 3 \leq 2^3 \).

Induction Hypothesis: We complete the proof using strong induction. The induction hypothesis is therefore as follows: For every \( n' \leq n \) the statement \( s_{n'} \leq 2^{n'} \) holds.

Induction Step: We need to show that the statement also holds for \( n + 1 \):

\[
s_{n+1} = s_n + s_{n-1} + s_{n-2} \leq 2^n + 2^{n-1} + 2^{n-2} = 2^{n-2}(4 + 2 + 1) \leq 2^{n-2} \cdot 8 = 2^{n+1}.
\]

\[\checkmark\]

2 Loop Invariant

Prove that the stated invariant holds throughout the execution of the loop (using the Initialization, Maintenance, Termination approach discussed in the lectures):

Algorithm 1 Algorithm \( \mathcal{A} \)

**Require:** Array \( A \) of length \( n \) (\( n \geq 2 \))

2: for \( i \leftarrow 1 \ldots n - 2 \) do
3: \( S \leftarrow S + A[i] - A[i+1] \)
4: end for
5: return \( S \)

**Invariant:**

At the beginning of iteration \( i \), \( S = A[0] - A[i] \) holds.

What does the algorithm compute?
Solution. Let $S_i$ be the value of $S$ at the beginning of iteration $i$.


3. **Termination**: We have that after the last iteration (or before the $n-1$th iteration that is never executed) $S = A[0] - A[n-1]$. The algorithm thus computes the value $A[0] - A[n-1]$.

3 Insertionsort

What is the runtime (in $\Theta$-notation) of Insertionsort when executed on the following arrays of lengths $n$:

1. $1, 2, 3, 4, \ldots, n-1, n$

**Solution.** The runtime is $\Theta(n)$ since the inner loop of Insertionsort always requires time $\Theta(1)$ on this instance (no moves are needed).

2. $n, n-1, n-2, \ldots, 2, 1$

**Solution.** The runtime is $\Theta(n^2)$. An easy way to see this is as follows: Consider the last $n/2$ elements of the input array. Each of these elements is moved at least $n/2$ positions to the left, i.e., the inner loop requires time $\Theta(n)$ for each of these elements. The total runtime is therefore $\Omega(n \cdot \frac{n}{2}) = \Omega(n^2)$. Since the runtime of Insertionsort is $O(n^2)$ on any instance, the runtime has to be $\Theta(n^2)$.

3. The array $A$ such that $A[i] = 1$ if $i \in \{1, 2, 4, 8, 16, \ldots\}$ (i.e., when $i$ is a power of two) and $A[i] = i$ otherwise.

**Solution.** Observe that Insertionsort does not move any of the elements (i.e., executes the inner loop) that are outside the positions $i \in \{1, 2, 4, 8, 16, \ldots\}$. We thus only need to count the number of iterations of the inner loop for these positions. Observe further that the element at position $2^j$, for some integer $j$, is moved at most $2^j$ steps to the left. Furthermore, we have that $2^{\lceil \log n \rceil} \geq 2^{\log n} = n$. Hence, there are at most $\lceil \log n \rceil$ positions in $A$ with value 1. The total number of iterations the inner loop of Insertionsort is executed is therefore at most:

$$\sum_{j=0}^{\lceil \log n \rceil} 2^j = 2^{\lceil \log n \rceil+1} - 1 \leq 2^{\log n+2} - 1 = 4n - 1 = \Theta(n).$$

Here we used the inequality $\lceil \log n \rceil \leq \log(n) + 1$, and the formula $\sum_{j=0}^{k} 2^j = 2^{k+1} - 1$. The runtime therefore is $\Theta(n)$. 

✓
4 Runtime Analysis

Algorithm 2

Require: Integer $n \geq 2$

\begin{verbatim}
x ← 0
i ← n
while $i \geq 2$ do
  $j ← \lceil n^{1/4} \rceil \cdot i$
  while $j \geq i$ do
    $x ← x + 1$
    $j ← j - 10$
  end while
  $i ← \lfloor i/\sqrt{n} \rfloor$
end while
return $x$
\end{verbatim}

Determine the runtime of Algorithm 3 in $\Theta$-notation.

Solution. Let us first determine the number of times $x$ the inner loop is executed. The value of $j$ evolves as follows:

\[ n^{1/4} \cdot i, n^{1/4} \cdot i - 10, n^{1/4} \cdot i - 20, \ldots \]

until it reaches a value that is smaller than $i$. We thus have $n^{1/4} \cdot i - x \cdot 10 < i$ which yields $x = \Theta(n^{1/4} i)$.

Next, concerning the outer loop, we see that the parameter $i$ evolves as follows (disregarding the floor operation): $n, n/\sqrt{n} = \sqrt{n}, 1$. In fact, the iteration with $i = 1$ is never executed. The overall runtime therefore is:

\[ \Theta(n^{1/4} n) + \Theta(n^{1/4} \sqrt{n}) = \Theta(n^{5/4}) \]

i.e., the runtime is dominated by the first iteration of the outer loop. \(\checkmark\)