Theory I, Sheet 0

Exercise 0.1 - Proof by induction [Points: *]

1. Prove by induction that
$$\sum_{i=0}^{n} i^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$$
for any $n \in \mathbb{N}$.

2. Consider the definition of a binary tree. Either, a tree is

   (a) a leaf, written $\square$,

   (b) or an inner node with two children t_1 and t_2, which are both binary trees, written $N(t_1, t_2)$.

   The number of inner nodes of a binary tree $I(t)$ is defined as:
$$I(t) = \begin{cases} 0 & \text{if } t = \square \\ I(t_1) + I(t_2) + 1 & \text{if } t = N(t_1, t_2) \end{cases}$$
   The number of leafs of a tree t is defined by
$$L(t) = \begin{cases} 1 & \text{if } t = \square \\ L(t_1) + L(t_2) & \text{if } t = N(t_1, t_2) \end{cases}$$
   Prove that the difference between the number of leaves and the number of internal nodes in a binary tree is 1.

Exercise 0.2 - Complexity [Points: *]

Characterize the relationship between $f(n)$ and $g(n)$ in the following examples using the $\mathcal{O}$-, Θ- or Ω-notation.

1. $f(n) = n^{0.99998}$ $\quad g(n) = \sqrt{n}$
2. $f(n) = 2^{\log^2(n)}$ $\quad g(n) = \sum_{k=1}^{n^2} \frac{n}{2^k}$
3. $f(n) = n \log_2(n)$ $\quad g(n) = \sqrt[3]{n}$
4. $f(n) = \sqrt{n}$ $\quad g(n) = 1000n$

Exercise 0.3 - Complexity [Points: *]

In order to solve a certain problem, five different algorithms $A_1, \ldots, A_5$ were developed. Algorithm A_i needs $T_i(n)$ time steps to solve the problem for an instance of size n.

1. $T_1(n) = 1000n$
2. $T_2(n) = 500n \log_2(n)$
3. $T_3(n) = n\sqrt{n}$
4. $T_4(n) = 10n^3$

5. $T_5(n) = 2^n$

The algorithms will be executed on a Pentium 1GHz processor. For simplicity, we assume that the processor executes exactly 10^9 computations per second.
Compute for any i the input size n, for which the problem can be solved by algorithm A_i within 1h.

Exercise 0.4 - Complexity classes [Points: *]

Given the following classes, DLOG, PSPACE, PTIME, NP, coNP, NLOG order them by set inclusions. (e.g. NLOG $\subseteq$ PTIME)