Algorithms and Data Structures

Homework 2, Problem 3

First name and last name: Göksenin Cakir

Proof of transitivity

$f \textcircled{\leq} g$ means $f \in O(g)$

Claim: $f(n) \in O(g(n))$ and $g(n) \in O(h(n)) \Rightarrow f(n) \in O(h(n))$

Proof:

$f(n) \in O(g(n))$ means: there are constants $n_1 > 0$ and $c_1 > 0$ such that $0 \leq f(n) \leq c_1 g(n)$ for $n \geq n_1$

$g(n) \in O(h(n))$ means: there are constants $n_2 > 0$ and $c_2 > 0$ such that $0 \leq g(n) \leq c_2 h(n)$ for $n \geq n_2$.

We obtain $0 \leq f(n) \leq c_1 g(n) \leq c_1 c_2 h(n)$ for $n \geq \max(n_1, n_2)$.

Since $c = c_1 c_2$ and $n_0 = \max(n_1, n_2)$ are positive constants, $f(n) \in O(h(n))$ follows by the O definition.

Proof of reflexivity

We want to show $f(n) \in O(f(n))$, which means: there are constants $n_0 > 0$, $c > 0$ such that $0 \leq c f(n) \leq f(n)$ for $n \geq n_0$.

This is true, because $1 \cdot f(n) \leq f(n)$ for all natural numbers, and $f(n)$ is assumed to be asymptotically nonnegative.

More precisely, $c = 1$ and some large enough n_0 satisfy the definition, so $f(n) \in O(f(n))$