This document is intended as a sample illustrating some techniques you might use in your CS252 problem set write-ups. But go ahead and read the content of this document, too.

1 Definitions

Definition 1.1 *Given two functions* $f, g : \mathbb{N} \to \mathbb{R}^+$*, we say that* $\mathbf{f} = \mathbf{O}(\mathbf{g})$ *if* $\exists c > 0, n_0 > 0$ *such that* $n \geq n_0 \implies f(n) \leq cg(n)$*.*

Definition 1.2 *Given two functions* $f, g : \mathbb{N} \to \mathbb{R}^+$*, we say that* $\mathbf{f} = \mathbf{\Omega}(\mathbf{g})$ *if* $\exists\ c > 0, n_0 > 0$ *such that* $n \geq n_0 \implies f(n) \geq cg(n)$*.*

Definition 1.3 *Given two functions* $f, g : \mathbb{N} \to \mathbb{R}^+$*, we say that* $\mathbf{f} = \mathbf{\Theta}(\mathbf{g})$ *if* $\exists\ c_1 > 0, c_2 > 0, n_0 > 0$ *such that* $n \geq n_0 \implies c_1 g(n) \leq f(n) \leq c_2 g(n)$*.*

2 A few notes about O, Ω, and Θ

- $O(f)$ is commonly pronounced "big oh of f", "oh of f", or "order f".
- There are several commonly-used variants of these notations. For example, a big-oh relationship can be denoted in any of these ways:
  - $f = O(g)$
  - $f(n) = O(g(n))$
  - f is $O(g)$
  - $f(n)$ is $O(g(n))$
  - $f \in O(g)$
  - $f(n) \in O(g(n))$
- Following the textbook's lead, we will use the = notation, either $f = O(g)$ or $f(n) = O(g(n))$ depending on context and personal preference.

3 Jeff's personal preference, and why we're ignoring it

I personally prefer the $\in$ style of notation, because it best fits the way I think about O and its friends. In my preferred notational approach, we think of $O(g)$ as a set of functions with a particular relationship to g, like so:

Alternative definition: Given a function $g : \mathbb{N} \rightarrow \mathbb{R}^+$, define $\mathbf{O(g)}$ to be the set of all functions $f : \mathbb{N} \rightarrow \mathbb{R}^+$ for which $\exists c > 0, n_0 > 0$ such that $n \geq n_0 \implies f(n) \leq cg(n)$.

This is the style of definition used by a very famous Algorithms book.

The reason I like this set-based way of thinking about O, Ω, and Θ is that it makes many important assertions simpler (at least they seem simpler to me). For example, if I say $O(n^2) = O(n^2 + n)$, I'm saying that two sets are equal, and that's all.

If you don't think about $O(g)$ as being a set, however, then the statement $O(n^2) = O(n^2+n)$ is a bit more cumbersome. It's either something intuitively correct but imprecise like "n^2 and $n^2 + n$ grow at the same rate", or something more precise but harder to say, like "for any function f, $f = O(n^2) \iff f = O(n^2 + n)$".

OK, so that's why I like the set-based $\in$-notation way of thinking about O, Ω, and Θ. But **our textbook most commonly uses $f(n) = O(g(n))$, so we'll do that, too**. I'll live.