CIS 121—Data Structures and Algorithms with Java - Fall 2015

Lab 2 — TBD

Learning Goals

During this lab, you will:

- review Bachmann-Landau notation
- examine certain functions and their relative asymptotic growth rates
- examine the runtime complexity of code
- prove Bachmann-Landau relations

Big-Oh and Bachmann-Landau Notation

In class, you have started to discuss Big Oh and other ways of classifying functions and algorithms. These notations belong to what is commonly referred to as the *Bachmann-Landau* family of notations.

Big-Oh Notation

Definition. $f(n) = O(g(n))$ if there exist constants n_0 and $c > 0$ s.t. $f(i) \leq cg(i)$ for all $i \geq n_0$.

Big-Omega Notation

Definition. $f(n) = \Omega(g(n))$ if there exist constants n_0 and $c > 0$ s.t. $f(i) \geq cg(i)$ for all $i \geq n_0$.

Big-Theta Notation

Definition. $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

As a protip, it is also good to note that the Bachmann-Landau notations refer to *classes of functions.* When you read $f(n) = O(g(n))$, this is equivalent to the statement:

$$f(n) \in O(g(n))$$

. Specifically, $f(n)$ is in the set of functions which are asymptotically bounded above by $g(n)$.