**Test One CS:4050**

Loop Invariant - A **loop invariant** is a condition [among program variables] that is necessarily true immediately before and immediately after each iteration of a **loop**. (Note that this says nothing about its truth or falsity part way through an iteration.)

**Three necessary Conditions of a Loop Invariant:**

* **Initialization**
* **Maintenance**
* **Termination**

**Algorithmic Notation**

**Rules of Logs**

1) *log*b(*mn*) = *log*b(*m*) + *log*b(*n*)

2) *log*b(*m*/*n*) = *log*b(*m*) – *log*b(*n*)

3) *log*b(*mn*) = *n* · *log*b(*m*)

<https://www.cs.auckland.ac.nz/courses/compsci220s1c/lectures/2014S1C/Part1/220-03.pdf>

**Big O-** By definition, g(n) is O(f(n)), or g(n) = O(f(n)) if a constant

c > 0 exists, such that cf(n) grows faster than g(n) for all n > n0.

**Big Omega-** iff there exists a positive real constant c and a positive integer n0

such that g(n) ≥ cf(n) for all n > n0.

**Big Theta-** iff there exists two positive real constants c1 and c2 and a positive

integer n0 such that c1f(n) ≤ g(n) ≤ c2f(n) for all n > n0.

**Order of Function Growth**

1. Exponential
2. Polynomial
3. Polylogrithmic
4. Logrithmic
5. Constant

**Recurrence Relations**

**Iteration**

**Prove:**

**Master Method**

Case 1:

Case 2:

Case 3:

Regularity Condition: