Section 1

Log Identities

Recall $k = \log(x) = \log_2(x) \Leftrightarrow 2^k = x$

- $\log(x*y) = \log(x) + \log(y)$
- $\log(x/y) = \log(x) - \log(y)$
- $\log(x^d) = d*\log(x)$
- $x^{\log(y)} = y^{\log(x)}$
- Change of base formula: $\log_a x = \log_b(x) / \log_b(a)$

Proof methods: $n^2 \geq n$ for all $n \geq 1$

Directly:	**By contradiction**:	**By induction**:
$n \geq 1$ $n*n \geq n*1$ $n^2 \geq n$	Assume to the contrary there exists $k \geq 1$ such that $k^2 < k$. But then $k^2 < k$ $k^2/k < k/k$ $k < 1$ Contradiction.	● Base case: $1^2 = 1 \geq 1$ ● Inductive step: assume $(k-1)^2 \geq k-1$, then $k^2 = (k-1)^2 + 2k - 1$ $\geq k - 1 + 2k - 1$ $\geq k$ ● Therefore $n^2 \geq n$ for all $n \geq 1$

Big-O, Big-Omega, Big-Theta

- $T(n) = O(g(n)) \Leftrightarrow \exists\ c, n_0$ s.t. $\forall\ n \geq n_0,\ 0 \leq T(n) \leq c*g(n)$
- $T(n) = \Omega(g(n)) \Leftrightarrow \exists\ c, n_0$ s.t. $\forall\ n \geq n_0,\ 0 \leq c*g(n) \leq T(n)$
- $T(n) = \Theta(g(n)) \Leftrightarrow T(n) = O(g(n))$ **and** $T(n) = \Omega(g(n))$

Master Theorem

If $T(n) = a*T(n/b) + O(n^d)$ with $a \geq 1$, $b > 1$, and $T(1) = O(1)$, then

- $T(n) = O(n^d \log(n))$ if $a = b^d$
- $T(n) = O(n^d)$ if $a < b^d$
- $T(n) = O(n^{log\_b(a)})$ if $a > b^d$