COMS 311 Equation Sheet

Notations. Definitions for the various notations for easy access while studying

Big-O: $O(g(n)) = \{f(n) : \exists c > 0, \exists n_0 > 0$ such that $0 \leq f(n) \leq cg(n), \forall n \geq n_0\}$

- g(n) is a tight upper bound for f(n)

Big-Ω: $\Omega(g(n)) = \{f(n) : \exists c > 0, \exists n_0 > 0$ such that $0 \leq cg(n) \leq f(n), \forall n \geq n_0\}$

- g(n) is a tight lower bound for f(n)

Big-Θ: $\Theta(g(n)) = \{f(n) : 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0$ for positive constants c_1, c_2, and $n_0\}$

- $f(n) \in O(g(n)) \bigwedge f(n) \in \Omega(g(n)) \to f(n) \in \Theta(g(n))$
- Basically if f(n) is both in O(g(n)) and Ω(g(n)), then it is in Θ(g(n))

Little-o: $o(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0$ such that $0 \leq f(n) < cg(n), \forall n \geq n_0\}$

- This one is tricky, since we want strictly greater than, we need to establish a g(n) such that any choice of c greater than 0 is strictly greater than f(n)

Little-ω: $\omega(g(n)) = \{f(n) : \forall c > 0, \exists n_0 > 0$ such that $0 \leq cg(n) < f(n), \forall n \geq n_0\}$

- Similar idea as little-o, but in this instance, we want g(n) to be strictly less than f(n) for any choice of c greater than 0
- Shortcut: f(n) $\in \omega$(g(n)) iff g(n) $\in$ o(f(n))

Equalities. Use these for simplifying various summations

$\sum_{i=1}^{n} i = n(n+1)/2$ $\qquad$ $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$

Finite Series. Use these with Recurrence Trees for recognizing the relationship

$a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} = a\frac{1-(r^n)}{(1-r)}$

$1 + 2 + 4 + \cdots + 2^n = 2^{n+1} - 1$

$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$

$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$

$1^3 + 2^3 + 3^3 + \cdots + n^3 = \frac{n^2(n+1)^2}{2^2}$

$1 + r + r^2 + r^3 + \cdots + r^{n-1} = \frac{1-r^n}{1-r}$, if $r \neq 1$

Infinite Series. Also for recognizing patterns

$1 + r^2 + r^3 + \cdots = 1/(r-1)$ when r < 1

$1 + 1/r + 1/r^2 + 1/r^3 + \cdots = \frac{1}{1-1/r}$ (when r < 1)

$\sum_{0}^{logn} kx^k = \frac{x}{(1-x)^2}$

Log Properties. Use these to break logs down into more useful terms

$a^x = y \Rightarrow log_a(y) = x$	$log_c(ab) = log_c(a) + log_c(b)$	$log_b(a) = \frac{1}{log_a(b)}$
$y = a^x = a^{log_a(y)}$	$log_c(a/b) = log_c(a) - log_c(b)$	$log_c(1/a) = -log_c(a)$
$a^{log_b(c)} = c^{log_b(a)}$	$log_c(a^n) = nlog_c(a)$	$*log_b(a) = \frac{log_c(a)}{log_c(b)}$

Master's Theorem. When of the form T(n)=aT($\lfloor n/b \rfloor$) + $f(n)$, then there are three cases to know

Case 1. $f(n) = O(n^{log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) \in \Theta(n^{log_b a})$

Case 2. $f(n) = \Theta(n^{log_b a} lg^k n)$ for some constant $k \geq 0$, then $T(n) \in \Theta(n^{log_b a} lg^{k+1} n)$

Case 3. $f(n) = \Omega(n^{log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(\lceil n/b \rceil) \leq cf(n)$ (or $af(\lfloor n/b \rfloor) \leq cf(n)$) for some constant c < 1 and all sufficiently large n, then $T(n) \in \Theta(f(n))$

Substitution Method. This method is used with induction to form a proof of a recurrence relation

In my opinion strong induction is best used here, by assuming our invariant holds for m and all values less than m, is invaluable for the proof. The substitution comes in two ways:

Substitution 1. This is the main/official substitution that gives it the name

Since we assume that $T(m) \in O(g(n))$, then $\exists c > 0$ such that $T(m) \leq cg(n)$, and thus we can replace T(m) with cg(n). This substitution is the main idea of the method.

Substitution 2. This is the secondary substitution, that is important, but not technically a part of the method itself.

Basically in a similar idea to substitution 1 above, since $\lfloor n/b \rfloor \leq (n/b)$, then we can also just replace $\lfloor n/b \rfloor$ with (n/b). There is a way to work with ceiling that was covered in lecture, but it is much more complicated and you most likely wont need to know that for this class.