COMP 3711H: Mathematical Facts for Midterm – *Version of October 15, 2016*

Graphs: A graph G is often denoted as $G = (V, E)$ where E is the set of its *vertices* and $E \subseteq V \times V$ is the set of its *edges.* In a *directed graph*, edges have *direction*, i.e, (v_1, v_2) denotes an edge pointing from vertex v_1 to vertex v_2 and $(v_1, v_2) \neq (v_2, v_1)$. In an *undirected graph*, edge (v_1, v_2) denotes the same edge as (v_2, v_1).

A graph $G = (V, E)$ is *complete* if, for all vertices $u, v \in V$, $u \neq v$, the edge (u, v) is in E. Thus, a complete undirected graph has $|V|(|V| - 1)/2$ edges while a complete directed graph has $|V|(|V| - 1)$ edges.

$$P = \{(v_1, v_2),\ (v_2, v_3),\ (v_3, v_4), \ldots,\ (v_{k-1}, v_k)\}$$

is a *path* connecting vertices u and v if $v_1 = u$, and $v_k = v$.

A *tree* is a directed acyclic graph. A tree with $|V|$ vertices has $|V| - 1$ edges.

A binary tree of height h has at most 2^h leaves.

Asymptotic Forms: The following gives both the formal "c and n_0" definitions and an equivalent limit definition for the standard asymptotic forms. Assume that f and g are non-negative functions. Usually f takes the role of the running time of an algorithm that we wish to analyze, and g takes the form of the asymptotic function to which we wish to compare f.

Asymptotic Form	Limit Form	Formal Definition
$f(n) = O(g(n))$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$	$\exists c, n_0, \forall n \geq n_0,\ f(n) \leq cg(n)$
$f(n) = \Omega(g(n))$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0$	$\exists c, n_0, \forall n \geq n_0,\ cg(n) \leq f(n)$
$f(n) = \Theta(g(n))$	$0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$	$f = O(g(n))$ and $f \in \Omega(g(n))$

Common Log Identities: The following are useful in simplifying asymptotic expressions involving logs. Let a, b, and c be positive constants. We use lg to denote $\log_2$ and ln to denote the natural log. When the base does not matter (as in asymptotic expressions) we just use log.

$$\begin{aligned}
\log(a \cdot b) &= \log a + \log b \\
\log(a^b) &= b \log a \\
a^{\log_a b} &= b \\
a^{\log_b c} &= c^{\log_b a} \\
\log_a n &= \frac{\log_b n}{\log_b a} = \Theta(\log n) \\
\log(n!) &= \Theta(n \log n)
\end{aligned}$$

Common Summations: Let $c \neq 1$ be any positive constant and assume $n \geq 0$. The following are the most common summations that arise when analyzing algorithms and data structures. You should memorize their asymptotic values.

Name of Series	Formula	Closed-Form Solution	Asymptotic Form
Constant	$\sum_{i=1}^{n} 1$	$= n$	$\Theta(n)$
Arithmetic	$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n$	$= \frac{n(n+1)}{2}$	$\Theta(n^2)$
Polynomial	$\sum_{i=1}^{n} i^c = 1^c + 2^c + \cdots + n^c$	(none for general c)	$\Theta(n^{c+1})$
Geometric	$\sum_{i=0}^{n-1} c^i = 1 + c + c^2 + \cdots + c^{n-1}$	$= \frac{c^n - 1}{c - 1}$	$\Theta(c^n)$ $(c > 1)$ $\Theta(1)$ $(c < 1)$
Harmonic	$\sum_{i=1}^{n} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$	$= \ln n + O(1)$	$\Theta(\log n)$

(Simplified) Master Theorem for Recurrences: This is very useful when dealing with recurrences arising from divide-and-conquer algorithms. Let $a \geq 1$, $b > 1, c \geq 0$ be constants. If $T(n)$ is the recurrence $T(n) = aT(n/b) + \Theta(n^c)$, defined for $n \geq 1$.

Case 1: $c < \log_b a$ then $T(n)$ is $\Theta(n^{\log_b a})$.

Case 2: $c = \log_b a$ then $T(n)$ is $\Theta(n^c \log n)$.

Case 3: $c > \log_b a$ then $T(n)$ is $\Theta(n^c)$.

If instead $T(n)$ is the recurrence *inequality* defined by $T(n) \leq aT(n/b) + O(n^c)$, for $n \geq 1$ then

Case 1: $c < \log_b a$ then $T(n)$ is $O(n^{\log_b a})$.

Case 2: $c = \log_b a$ then $T(n)$ is $O(n^c \log n)$.

Case 3: $c > \log_b a$ then $T(n)$ is $O(n^c)$.

Other common recurrences: Let $b > 1, c$ be any constants.

$T(n) = T(n/b) + \Theta(c) \quad \Rightarrow \quad T(n) = \Theta(\log n).$

$T(n) = bT(n/b) + \Theta(c) \quad \Rightarrow \quad T(n) = \Theta(n).$

and

$T(n) \leq T(n/b) + O(c) \quad \Rightarrow \quad T(n) = O(\log n).$

$T(n) \leq bT(n/b) + O(c) \quad \Rightarrow \quad T(n) = O(n).$

Note: $\Theta(c) = \Theta(1)$ and $O(c) = O(1)$ for all constants $c > 0$. Recall that $\Theta(1)$ means a term that's bounded from both above and below by some constants greater than 0. In particular, it can't be a term that's decreasing to zero. $O(n)$ means a term that is bounded from above by a constant. It *can* (but doesn't have to be) a term that is decreasing to zero.