COMP 3711:Spring 2019

COMP 3711: Exam Math Handout

Common Log Identities:

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

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.

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 and let $T(n)$ be the recurrence $T(n) = aT(n/b) + n^c$, 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 $T(n) \le aT(n/b) + O(n^c)$, for $n \ge 1$ 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

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

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

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.

Probabilistic Statements

1. Expectation: the expectation of discrete random variable X is
$$E(X) = \sum_i i \cdot \Pr(X = i).$$
2. Linearity of Independence: given two random variables X and Y (not necessarily independent),
$$E(X + Y) = E(X) + E(Y).$$
3. Indicator Random Variables: if X is a random variable that takes on only values 0 and 1 then
$$E(X) = \Pr(X = 1).$$
4. Waiting time for first success: a coin comes up heads with probability p and tails with probability $1 - p$. If X is the random variable counting the number of coin flips made until a head comes up the first time then
$$E(X) = \frac{1}{p}.$$

Tree Facts

1. A heap of height h has between 2^h and $2^{h+1} - 1$ nodes.
2. A binary tree of height h has at most 2^h leaves.