CS 313, Intermediate Data Structures

Logarithm Cheat Sheet

Definition: $m = \log_a n$ if and only if $n = a^m$, where $a > 0$ (thus $n > 0$).

In other words, logarithms are exponents.

a is called the base of the algorithm.

Example: $\log_2 8 = \log_2 2^3 = 3$ as $2^3 = 8$.

In order for $\log_a n$ to be valid, both a and n have to be strictly greater than 0.

In this class, n is often the size of the input, so it's often a positive integer, thus satisfying the requirement to take logarithms.

Properties of Logarithms

For all $a > 0$, $b > 0$, $n > 0$ and $m > 0$:

1. $\log_a a = 1$
2. $\log_a 1 = 0$
3. $\log_a nm = \log_a n + \log_a m$
4. $\log_a \frac{n}{m} = \log_a n - \log_a m$
5. $\log_a n^m = m \log_a n$
6. $\log_a a^n = n$
7. $a^{\log_a n} = n$
8. $\log_a n = \frac{\log_b n}{\log_b a}$ (taking $\log_b$ on both sides of property 7. and applying property 5.)

Remarks:

- Because of property 8., the base of the logarithms does not matter when we apply asymptotic notations
- When the base $a = e \approx 2.71828$ (i.e, the Euler number), we often write $\log_a n = \ln n$ (called the natural logarithm)

- As the base doesn't matter, we often use the base of 2 ($a = 2$) in the asymptotic notations. So, we often simply write $\log n$ or sometime lg n to imply $\log_2 n$ in class (i.e., $\log n = \lg n = \log_2 n$).
- In general, the logarithm functions grow much slower than the polynomial functions that are in turn, grow much slower than the exponential functions, (in the o notation: $\log^k n = o(n^\alpha) = o(c^n)$ for all $k > 0, \alpha > 0, c > 1$).