Asymptotic Notation

- $f(n) = o(g(n)):\ 0 \leq f(n) < cg(n))$
- $f(n) = O(g(n)):\ 0 \leq f(n) \leq cg(n)$ (upper bound)
- $f(n) = \Theta(g(n)) \leftrightarrow \Omega(g(n))$ & $O(g(n))$
- $f(n) = \Omega(g(n):\ 0 \leq cg(n) \leq f(n)$ (lower bound)
- $f(n) = \omega(g(n):\ 0 \leq cg(n) < f(n)$

Useful Equations

$$\sum_{k=1}^{n} k = \frac{1}{2}n(n+1) = \Theta(n^2)$$

Amortized Analysis

$$\begin{aligned} \hat{c}_i &= c_i + \Phi(D_i) - \Phi(D_{i-1}) \\ \sum_{i=1}^{n} \hat{c}_i &= \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0) \end{aligned}$$

Master Method For recurrences of the form $T(n) = aT(n/b) + f(n)$, where a is the number of branches at each level, b is the reduction in size at each level, and $f(n)$ is the initial cost:

Case 1: $f(n) = O(n^{\log_b a - \epsilon}) \rightarrow T(n) = \Theta(n \log_b a), \epsilon > 0$: $f(n)$ is polynomially slower than $n^{\log_b a}$, weight increases toward leaves because $b < a$

Case 2: $f(n) = \Theta(n^{\log_b a}) \rightarrow T(n) = \Theta(n^{\log_b a} \log^{k+1} n), k \geq 0$: $f(n)$ & $n^{\log_b a}$ grow similarly, weight remains constant because $b = a$

Case 3: $f(n) = \Omega(n^{\log_b a + \epsilon}) \rightarrow T(n) = \Theta(f(n))$: weight decreases because $b > a$

Recursion Trees