This document contains notes on important takeaways from COMP160.

1 Time Complexity

2 Master Method

Simple

Given a recurrence $T(n) = aT(\frac{n}{b}) + \Theta(n^d)$

- $\log_b a > d \implies T(n) = \Theta(n^{\log_b a})$

  This is the case where the leaves dominate the asymptotic growth.

- $\log_b a = d \implies T(n) = \Theta(n^d \log n)$

  This is the case where each level donates equally to the asymptotic growth.

- $\log_b a < d \implies T(n) = \Theta(n^d)$

  This is the case where the work done at the root of the recursion tree is the greatest.

Complex

Given a recurrence $T(n) = aT(\frac{n}{b}) + f(n)$ and $a, b \in \mathbb{R}$ s.t. $a > 0, b > 1$

- $f(n) = O(n^{\log_b a - \epsilon})$, for some $\epsilon > 0 \implies T(n) = \Theta(n^{\log_b a})$

- $f(n) = \Theta(n^{\log_b a}) \implies T(n) = \Theta(n^{\log_b a} \log n)$

  Also, $f(n) = \Theta(n^{\log_b a} \log^k n) \implies T(n) = \Theta(n^{\log_b a} \log n), k \geq 0 \implies T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$

- $f(n) = \Omega(n^{\log_b a + \epsilon})$, for some $\epsilon > 0 \implies T(n) = \Theta(f(n))$