CISC 3220 Homework Master Theorem

Rachel Friedman

March 1, 2020

Master Theorem

Recurrence relations that are of the form $aT(\frac{n}{b}) + f(n)$, where $a \geq 1$ and $b > 1$ can be solved with the Master Theorem. Establish which one of the three cases is applicable, and then select the appropriate runtime.

Case 1: $a > b^d$ (more work)
In this case, the work keeps increasing at each level of recursion and so the runtime is proportional and dominated by the work done at the leaves. This is the case when $f(n) = \mathcal{O}(n^{log_b a})$ or $0 \leq f(n) \leq n^{log_b a - \epsilon}$ for some constant $\epsilon > 0$, and so the runtime is $T(n) = \Theta(n^{log_b a})$.

Case 2: $a = b^d$ (equal work)
In this case, the work is evenly distributed. This is the case when $f(n) = \Theta(n^{log_b a})$, or $c_1 \cdot n^{log_b a} \leq f(n) \leq c_2 \cdot n^{log_b a}$. Then the runtime is $T(n) = \Theta(n^{log_b a} lgn)$.

Case 3: $a < b^d$ (less work)
In this case most of the work keeps decreasing at each level of recursion, so most of the work is done at the root. This is the case when $f(n) = \Omega(n^{log_b a + \epsilon})$ for some constant $\epsilon > 0$, or $0 \leq f(n) \geq n^{log_b a}$. It must also be the case that $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$ for some constant $c < 1$. Then the runtime is $T(n) = \Theta((f(n))$.

Intuitively, it is the largest function that determines the runtime.

Applying the Master Theorem

Each of the functions below apply to the cases above, respectively. Work it out!

$$T(n) = 9T(n/3) + n \tag{1}$$

$$T(n) = T(2n/3) + 1 \tag{2}$$

$$T(n) = 3T(n/4) + n\,lgn \tag{3}$$