CSC721 Algorithms Fall 2017

Master Theorem(s) a.k.a. "Somewhat Convenient Theorems"

Simple Version

The recurrence equation:

$$\begin{aligned} T(1) &= b \\ T(n) &= aT(n/c) + bn \end{aligned}$$

has the following solution:

$$T(n) \in \begin{cases} \Theta(n) & \text{if } a < c \\ \Theta(n \log n) & \text{if } a = c \\ \Theta(n^{\log_c a}) & \text{if } a > c \end{cases}$$

Somewhat Enhanced Version

The recurrence equation:

$$\begin{aligned} T(1) &= b \\ T(n) &= aT(n/c) + bn^d \end{aligned}$$

has the following solution:

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } \log_c(a) < d \\ \Theta(n^d \log n) & \text{if } \log_c(a) = d \\ \Theta(n^{\log_c a}) & \text{if } \log_c(a) > d \end{cases}$$

Advanced Version

The recurrence equation:

$$\begin{aligned} T(1) &= 1 \\ T(n) &= aT(n/c) + f(n) \end{aligned}$$

has solution:

$$T(n) \in \begin{cases} \Theta(n^{\log_c a}) & \text{if } f(n) \in \mathcal{O}(n^{\log_c(a)-\epsilon}) \text{ for some } \epsilon > 0 \\ \Theta(n^{\log_c a} \log^{k+1}(n)) & \text{if } f(n) \in \Theta(n^{\log_c(a)} \log^k(n)) \text{ where } k \geq 0 \\ \Theta(f(n)) & \text{if } f(n) \in \Omega\left(n^{\log_c(a)+\epsilon}\right) \text{ for some constant } \epsilon > 0 \text{ and} \\ & f\left(\frac{n}{c}\right) \leq \alpha f(n) \text{ for some constant } \alpha < 1 \text{ and sufficiently large } n \end{cases}$$