MCS-375: Algorithms: Analysis and Design Handout #DC1
San Skulrattanakulchai
Gustavus Adolphus College December 2, 2013

The Master Theorem

CLRS Ch 4

1. Divide-and-conquer recurrences

Suppose a divide-and-conquer algorithm divides the given problem into equal-sized subproblems, say a subproblems, each of size n/b. Its worst-case running time $T(n)$ then satisfies the recurrence

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ aT(n/b) + D(n) & \text{if } n > 1,\ n \text{ a power of } b. \end{cases}$$

Assume that a and b are real numbers, with $a > 0$ and $b > 1$.

Remarks.

1. Usually a in integral

2. Fractional b is useful, e.g., $T(n) = 3T(2n/3) + 1$. Here T is defined on a set of rational numbers $(3/2)^i$. The related function on integers, $T(n) = 3T(\lceil 2n/3 \rceil) + 1$, turns out to have the same asymptotic bound.

2. Solving the recurrences

Let $n = b^k$ so that $k = \log_b n$ (n is not necessarily integral but it is a power of b). Iterating the recurrence yields

$$\begin{aligned} T(n) = T(b^k) &= D(b^k) + aT(b^{k-1}) \\ &= D(b^k) + aD(b^{k-1}) + a^2T(b^{k-2}) \\ &= \sum_{i=0}^{k-1} a^i D(b^{k-i}) + a^k T(1). \end{aligned}$$

The second term $a^kT(1)$ is the solution when $D(\cdot) = 0$, called the *homogeneous solution (h.s.)*

$a^kT(1) = a^{\log_b n} = n^{\log_b a}$.

Let $h = \log_b a$ so h.s.$=n^h$.

Usually $h \geq 0$ since $a \geq 1$.

An Important Special Case

A common driving function is $D(n) = n^d$ where $d \geq 0$. The sum becomes $n^d \sum_{i=0}^{k-1} (a/b^d)^i$, a geometric progression.

Sum of a geometric progression:

$$\sum_{i=0}^{k-1} r^i = \begin{cases} \frac{r^k - 1}{r - 1} & \text{if } r \neq 1 \\ k & \text{if } r = 1. \end{cases}$$

Therefore, asymptotically

$$\sum_{i=0}^{k-1} r^i = \begin{cases} \Theta(1) & \text{if } 0 < r < 1 \\ \Theta(k) & \text{if } r = 1 \\ \Theta(r^k) & \text{if } r > 1. \end{cases}$$

Theorem. For $D(n) = n^d$,

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d, \text{ i.e., } h < d \\ \Theta(n^h \log n) & \text{if } a = b^d, \text{ i.e., } h = d \\ \Theta(n^h) & \text{if } a > b^d, \text{ i.e., } h > d. \end{cases}$$

Remark. Informally, "$T(n) = \max\{\text{homegeneous solution, driver}\}$"

Example.

$T(n) = 3T(2n/3) + 1.$

h.s. $= n^h$, $h = \log_{3/2} 3 \approx 2.7.$

$h > d$ $(\log_{3/2} 3 > 0) \Longrightarrow T(n) = \text{h.s.} = \Theta(n^h) = \omega(n^2).$

Exercises.

Give big-theta estimates of the following functions.

(i) $T(n) = T(n/2) + 1$

(ii) $T(n) = T(n/2) + n$

(iii) $T(n) = 2T(n/2) + n$

(iv) $T(n) = 2T(n/2) + n^2$

(v) $T(n) = 4T(n/2) + n$

(vi) $T(n) = 3T(n/2) + n$