CMPS 101
Algorithms and Abstract Data Types

Recurrence Relations

Iteration Method

Recall the following example from the induction handout.

$$T(n) = \begin{cases} 0 & n = 1 \\ T(\lfloor n/2 \rfloor) + 1 & n \geq 2 \end{cases}$$

We begin by illustrating a solution technique called *iteration*, which consists of repeatedly substituting the recurrence into itself until a pattern emerges.

$$\begin{aligned} T(n) &= 1 + T(\lfloor n/2 \rfloor) \\ &= 1 + 1 + T(\lfloor \lfloor n/2 \rfloor / 2 \rfloor) = 2 + T(\lfloor n/2^2 \rfloor) \\ &= 2 + 1 + T(\lfloor \lfloor n/2^2 \rfloor / 2 \rfloor) = 3 + T(\lfloor n/2^3 \rfloor) \\ &\vdots \\ &= k + T(\lfloor n/2^k \rfloor) \end{aligned}$$

This process must terminate when the recursion depth k is at its maximum, i.e. when $\lfloor n/2^k \rfloor = 1$. To solve this equation for k in terms of n, we use the inequality definition of the floor function.

$$\begin{aligned} & 1 \leq n/2^k < 2 \\ \therefore \quad & 2^k \leq n < 2^{k+1} \\ \therefore \quad & k \leq \lg(n) < k+1 \\ \therefore \quad & k = \lfloor \lg(n) \rfloor \end{aligned}$$

Thus for the recursion depth $k = \lfloor \lg(n) \rfloor$ we have $T(\lfloor n/2^k \rfloor) = T(1) = 0$, and hence the solution to the above recurrence is $T(n) = \lfloor \lg(n) \rfloor$. It follows that $T(n) = \Theta(\log(n))$.

Exercise
Check directly that $T(n) = \lfloor \lg(n) \rfloor$ is the solution to the above recurrence relation, i.e. check that $T(1) = 0$, and for any $n \geq 2$, that $T(n) = 1 + T(\lfloor n/2 \rfloor)$.

Exercise
Use this same technique to show that the recurrence

$$S(n) = \begin{cases} 0 & n = 1 \\ S(\lceil n/2 \rceil) + 1 & n \geq 2 \end{cases}$$

has solution $S(n) = \lceil \lg(n) \rceil$, and hence also $S(n) = \Theta(\log(n))$.

Comparing the solutions to the preceding examples, we see that replacing floor $\lfloor\ \rfloor$ by ceiling $\lceil\ \rceil$ has no affect on the *asymptotic solution* $\Theta(\log(n))$, while the *exact solutions* are different: $\lfloor \lg(n) \rfloor$ vs. $\lceil \lg(n) \rceil$. We can change other details of a recurrence without changing the asymptotic solution. The following recurrence satisfies $T(n) = \Theta(\log(n))$ for any values of the constants c, d, and n_0.

$$T(n) = \begin{cases} c & 1 \le n < n_0 \\ T(\lfloor n/2 \rfloor) + d & n \ge n_0 \end{cases}$$

Using the iteration method yields:

$$\begin{aligned} T(n) &= d + T(\lfloor n/2 \rfloor) \\ &= d + d + T(\lfloor n/2^2 \rfloor) \\ &\vdots \\ &= kd + T(\lfloor n/2^k \rfloor) \end{aligned}$$

We seek the first (i.e. the smallest) integer k such that $\lfloor n/2^k \rfloor < n_0$. This is equivalent to $n/2^k < n_0$, whence

$$\begin{aligned} &\frac{n}{n_0} < 2^k \\ &k-1 \le \lg(n/2^k) < k \qquad \text{(since } k \text{ is the least integer satisfying the previous inequality)} \\ &k-1 = \lfloor \lg(n/n_0) \rfloor \\ &k = \lfloor \lg(n/n_0) \rfloor + 1 \end{aligned}$$

Thus $T(n) = (\lfloor \lg(n/n_0) \rfloor + 1)d + c$ for $n \ge n_0$, and hence $T(n) = \Theta(\log(n))$ as claimed.

It is often difficult or impossible to determine an exact solution via the iteration method, while it *is* possible to obtain an asymptotic solution. Consider

$$T(n) = \begin{cases} 1 & n = 1 \\ T(\lfloor n/2 \rfloor) + n^2 & n \ge 2 \end{cases}$$

Upon iterating this recurrence we find $T(n) = \sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor^2 + 1$, where $k = \lfloor \lg n \rfloor$. We can use this expression to show that $T(n) = \Theta(n^2)$ as follows.

$$\begin{aligned} T(n) &= \sum_{i=0}^{k-1} \left\lfloor \frac{n}{2^i} \right\rfloor^2 + 1 \\ &\le n^2 \left(\sum_{i=0}^{k-1} (1/4)^i \right) + 1 \qquad \text{(since } \lfloor x \rfloor \le x \text{)} \end{aligned}$$

$$\leq n^2\left(\sum_{i=0}^{\infty}(1/4)^i\right) + 1 \qquad \text{(letting } k \to \infty\text{)}$$

$$= n^2\left(\frac{1}{1-(1/4)}\right) + 1 \qquad \text{(from a well known series formula)}$$

$$= \frac{4}{3}n^2 + 1$$

$$= O(n^2)$$

Therefore $T(n) = O(n^2)$. Showing that $T(n) = \Omega(n^2)$ is more difficult since estimating the floor function downward introduces some algebraic complexity.

$$T(n) = \sum_{i=0}^{k-1}\left\lfloor \frac{n}{2^i} \right\rfloor^2 + 1$$

$$\geq \sum_{i=0}^{k-1}\left(\frac{n}{2^i} - 1\right)^2 + 1 \qquad \text{(since } \lfloor x \rfloor > x - 1 \text{ for any } x\text{)}$$

$$= \sum_{i=0}^{k-1}\left(\frac{n^2}{4^i} - \frac{2n}{2^i} + 1\right) + 1$$

$$= n^2\left(\sum_{i=0}^{k-1}(1/4)^i\right) - 2n\left(\sum_{i=0}^{k-1}(1/2)^i\right) + k + 1$$

$$\geq n^2 - 2n\left(\frac{1-(1/2)^k}{1-(1/2)}\right) + k + 1 \qquad \text{(replacing the first sum by 1 and using a well known formula on the second)}$$

$$= n^2 - 4n\left(1 - \frac{1}{2^{\lfloor \lg n \rfloor}}\right) + \lfloor \lg n \rfloor + 1 \qquad \text{(using } k = \lfloor \lg n \rfloor\text{)}$$

$$\geq n^2 - 4n\left(1 - \frac{1}{2^{\lg n}}\right) + (\lg n - 1) + 1 \qquad \text{(since } \lfloor x \rfloor \leq x \text{ and } \lfloor x \rfloor > x - 1\text{)}$$

$$= n^2 - 4n + 4 + \lg n$$

$$= \Omega(n^2)$$

It follows that $T(n) = \Omega(n^2)$ and therefore $T(n) = \Theta(n^2)$, as claimed.

The Master Method

This is a method for finding (asymptotic) solutions to recurrences of the form $T(n) = aT(n/b) + f(n)$ where $a \geq 1$, $b > 1$, and the function $f(n)$ is asymptotically positive. Here $T(n/b)$ denotes either $T(\lfloor n/b \rfloor)$ or $T(\lceil n/b \rceil)$, and it is understood that $T(n) = \Theta(1)$ for some finite set of initial terms. Such a recurrence describes the run time of a 'divide and conquer' algorithm which divides a problem of size *n* into *a* subproblems, each of size n/b. In this context $f(n)$ represents the cost of doing the dividing and re-combining.

Master Theorem

Let $a \geq 1$, $b > 1$, $f(n)$ be asymptotically positive, and let $T(n)$ be defined by $T(n) = aT(n/b) + f(n)$. Then we have three cases:

(1) If $f(n) = O\left(n^{\log_b(a)-\varepsilon}\right)$ for some $\varepsilon > 0$, then $T(n) = \Theta\left(n^{\log_b(a)}\right)$.

(2) If $f(n) = \Theta\left(n^{\log_b(a)}\right)$, then $T(n) = \Theta\left(n^{\log_b(a)} \cdot \log(n)\right)$.

(3) If $f(n) = \Omega\left(n^{\log_b(a)+\varepsilon}\right)$ for some $\varepsilon > 0$, and if $af(n/b) \leq cf(n)$ for some $0 < c < 1$ and for all sufficiently large *n*, then $T(n) = \Theta\left(f(n)\right)$.

Remarks In each case we compare $f(n)$ to the polynomial $n^{\log_b(a)}$, then determine which of these two functions is of a higher asymptotic order. In case (1) $n^{\log_b(a)}$ is of higher order than $f(n)$ (by a polynomial factor n^{ε}) and the solution $T(n)$ is in the class $\Theta\left(n^{\log_b(a)}\right)$. In case (3) $f(n)$ is the higher order (again by a polynomial factor n^{ε}), and an additional *regularity condition* is satisfied. The Master Theorem tells us in this case that $T(n) = \Theta\left(f(n)\right)$. In case (2) the two functions are asymptotically equivalent, and $T(n)$ is in the class $\Theta\left(n^{\log_b(a)} \cdot \log(n)\right) = \Theta\left(f(n) \cdot \log(n)\right)$.

We sometimes say, as in case (1), that $n^{\log_b(a)}$ is *polynomially larger* than $f(n)$, meaning that $f(n)$ is bounded above by a function which is smaller than $n^{\log_b(a)}$ by a polynomial factor, namely n^{ε} for some $\varepsilon > 0$.

Example $T(n) = 8T(n/2) + n^3$

Observe that $a = 8$, $b = 2$, and $\log_b(a) = 3$. Hence $f(n) = n^3 = \Theta\left(n^{\log_b(a)}\right)$, so we are in case (2). Therefore $T(n) = \Theta(n^3 \log(n))$.

Example $T(n) = 5T(n/4) + n$.

In this case $a = 5$, $b = 4$, and $\log_b(a) = 1.1609... > 1$. Letting $\varepsilon = \log_4(5) - 1$ we have $\varepsilon > 0$, and therefore $f(n) = n = O\left(n^{\log_4(5)-\varepsilon}\right)$. Thus case(1) applies, and $T(n) = \Theta\left(n^{\log_4(5)}\right)$.

Example $T(n) = 5T(n/4) + n^2$

Again $a = 5$, $b = 4$, so that $\log_b(a) = 1.1609... < 2$. Upon setting $\varepsilon = 2 - \log_4(5)$, we have that $\varepsilon > 0$, and $f(n) = n^2 = \Omega\left(n^{\log_4(5)+\varepsilon}\right)$. If the Master Theorem applies at all then, it must be that case (3) applies. However, the regularity condition must also be checked: $5f(n/4) \leq cf(n)$ for some $0 < c < 1$ and all sufficiently large *n*. This inequality says $5(n/4)^2 \leq cn^2$, i.e. $(5/16)n^2 \leq cn^2$, which is true as long as *c* is chosen to satisfy $5/16 \leq c < 1$. By case (3) $T(n) = \Theta(n^2)$.

The conclusion reached by the Master Theorem does not change when we replace $f(n)$ by a function which is asymptotically equivalent to it. This is implied by the theorem since the hypothesis in each case refers only to the asymptotic growth rate of $f(n)$, not it's actual numerical values. Thus we can, if convenient, replace $f(n)$ by a any simpler asymptotically equivalent function. For instance the

recurrence $T(n) = 8T(n/2) + 10n^3 + 15n^2 - n^{1.5} + n\log(n) + 1$ can be reduced to $T(n) = 8T(n/2) + n^3$ which was our first example above, and we arrive at the very same asymptotic solution. (Of course the exact solution to the recurrence would be very different.) For this reason recurrences are sometimes specified in the form $T(n) = aT(n/b) + \Theta(f(n))$, if all that is required is an asymptotic solution. Notice that there is no mention of initial terms in the Master Theorem. It is part of the content of the theorem that the initial values of the recurrence do not effect it's asymptotic solution.

Observe that in the three preceding examples, $f(n)$ was a polynomial. This is a particularly easy setting in which to apply the Master Theorem, since to establish which case applies, one merely compares $\deg(f)$ to the number $\log_b(a)$. If they are the same, case (2) applies. If $\log_b(a)$ is larger, case (1) applies by defining $\varepsilon = \log_b(a) - \deg(f)$. When $\deg(f)$ is larger, case (3) applies upon setting $\varepsilon = \deg(f) - \log_b(a)$, and it turns out that the regularity condition automatically holds.

Exercise: Prove that if $f(n)$ is a polynomial, and if $\deg(f) > \log_b(a)$, then case (3) of the Master Theorem applies, and that the regularity condition necessarily holds.

Checking the hypotheses can be a little more complicated if $f(n)$ is not a polynomial.

Example $T(n) = T(\lfloor n/2 \rfloor) + 2T(\lceil n/2 \rceil) + \log(n!)$

First write this as $T(n) = 3T(n/2) + n\log(n)$. Let $\varepsilon = \frac{1}{2}(\log_2(3) - 1)$, then $\varepsilon > 0$, and $1 + \varepsilon = \log_2(3) - \varepsilon$.

Observe $n\log(n) = o(n^{1+\varepsilon})$, whence $n\log(n) = O(n^{1+\varepsilon}) = O(n^{\log_2(3)-\varepsilon})$. Case (1) gives $T(n) = \Theta(n^{\log_2(3)})$.

In spite of the name "Master Theorem" the three cases do not cover all possibilities. There is a gap between cases (1) and (2) when $n^{\log_b(a)}$ is larger than $f(n)$, but not *polynomially* larger. The following example illustrates this situation.

Example $T(n) = 2T(n/2) + n/\log(n)$.

Observe that $n/\log(n) = \omega(n^{1-\varepsilon})$ for any $\varepsilon > 0$, whence $n/\log(n) \neq O(n^{1-\varepsilon})$, and therefore we are not in case (1). But also $n/\log(n) = o(n)$, so that $n/\log(n) \neq \Theta(n)$, and neither are we in case (2). Thus the Master Theorem cannot be applied to this recurrence.

A similar gap exists between cases (2) and (3). It is also possible that the regularity condition in case (3) fails, even though $f(n) = \Omega(n^{\log_b(a)+\varepsilon})$ for some $\varepsilon > 0$.

Proof of the Master Theorem

We sketch here the proof of part (1) of the Master Theorem. Basically the proof is the iteration method applied with full generality. We will simplify matters by ignoring all floors and ceilings in the argument. Upon iteration we obtain

$$\begin{aligned} T(n) &= f(n) + aT(n/b) \\ &= f(n) + af(n/b) + a^2T(n/b^2) \\ &= f(n) + af(n/b) + a^2 f(n/b^2) + a^3T(n/b^3) \end{aligned}$$

$$\vdots$$
$$= \sum_{i=0}^{k-1} a^i f(n/b^i) \; + \; a^k T(n/b^k)$$

The recurrence terminates when $n/b^k = 1$, i.e. when the recursion depth k is $k = \log_b(n)$. For this value of k we have

$$T(n) \; = \; \sum_{i=0}^{k-1} a^i f(n/b^i) \; + \; a^{\log_b(n)} T(1) \; \geq \; \text{const} \cdot n^{\log_b(a)} \; = \; \Omega(n^{\log_b(a)})$$

It remains only to show $T(n) \leq O(n^{\log_b(a)})$, for then we have $T(n) = \Theta(n^{\log_b(a)})$ as required. Since we are in case (1) we have $f(n) = O\left(n^{\log_b(a)-\varepsilon}\right)$, and hence there exist positive numbers c and ε such that $f(n) \leq cn^{\log_b(a)-\varepsilon}$ for all sufficiently large n. Therefore if n is sufficiently large

$$\begin{aligned}
\sum_{i=0}^{k-1} a^i f(n/b^i) \; &\leq \; \sum_{i=0}^{k-1} a^i \cdot c\left(\frac{n}{b^i}\right)^{\log_b(a)-\varepsilon} \\
&= \; cn^{\log_b(a)-\varepsilon} \cdot \sum_{i=0}^{k-1} a^i \left(\frac{1}{b^i}\right)^{\log_b(a)-\varepsilon} \\
&= \; cn^{\log_b(a)-\varepsilon} \cdot \sum_{i=0}^{k-1} \frac{a^i}{(b^{\log_b(a)})^i} \cdot (b^\varepsilon)^i \\
&= \; \frac{cn^{\log_b(a)}}{n^\varepsilon} \cdot \sum_{i=0}^{k-1} (b^\varepsilon)^i \\
&= \; \frac{cn^{\log_b(a)}}{n^\varepsilon} \cdot \frac{(b^\varepsilon)^k - 1}{b-1} \\
&\leq \; \frac{c}{b-1} \cdot \frac{n^{\log_b(a)}}{n^\varepsilon} \cdot (b^{\log_b n})^\varepsilon \; = \; \frac{c}{b-1} \cdot n^{\log_b(a)},
\end{aligned}$$

whence $T(n) \leq \left(\frac{c}{b-1} + T(1)\right) n^{\log_b(a)} \; = \; \text{const} \cdot n^{\log_b(a)} \; = \; O(n^{\log_b(a)})$. ///

We emphasize that the above argument was only a "sketch" and not a complete proof, owing to the fact that we ignored floors and ceilings. The reader should fill in those details as an exercise. Also as an exercise, one should prove cases (2) and (3).