COS 423, February 13, 1996
Handout Number 3

About Recurrence Relations

In the analysis of algorithms, we often find the quantity under discussion, say $T(n)$, (such as the running time) satisfies some recurrence relation. How do we extract information about $T(n)$ from the recurrence relation? A general goal is to find a familiar function $g(n)$ (such as n^4 or $2^n \log n$) such that $T(n) = \Theta(g(n))$. We call this *determining $T(n)$ asymptotically up to a multiplicative factor.*

In class we used the recurrence relation arising from Strassen's matrix multiplication as an example. We summarize the discussions below. The general approach also applies to other recurrence relations as well.

Let $c > 0$ be some constant, and

$$\begin{aligned} T(1) &= 1, \\ T(n) &= 7T(\lceil n/2 \rceil) + cn^2 \quad \text{for } n > 1. \end{aligned} \tag{1}$$

Approach no. 1. See if there are some special values of n for which one can solve $T(n)$ exactly. For the present problem, the recurrence simplifies when n is restricted to be powers of 2, since we can then get rid of the ceiling function. We have for $n = 2^k$, by iterating the recurrence,

$$\begin{aligned} T(n) &= 7T(n/2) + cn^2 \\ &= 7(7T(n/2^2) + c(n/2)^2) + cn^2 \\ &= 7^2 T(n/2^2) + c(7(n/2)^2 + n^2) \\ &= \ldots \\ &= 7^k T(n/2^k) + c(7^{k-1}(n/2^{k-1})^2 + \cdots + 7^j (n/2^j)^2 + \cdots + 7(n/2)^2 + n^2) \\ &= 7^k + cn^2((7/4)^{k-1} + \cdots + (7/4)^j + \cdots + 7/4 + 1). \end{aligned} \tag{2}$$

Now $7^{\lg n} = (2^{\lg 7})^{\lg n} = (2^{\lg n})^{\lg 7} = n^{\lg 7}$, and

$$\begin{aligned} (7/4)^{k-1} + \cdots + (7/4)^j + \cdots + 7/4 + 1 &= ((7/4)^k - 1)/(7/4 - 1) \\ &= (3/7)(2^{k \lg(7/4)} - 1) \\ &= (3/7)(n^{\lg 7 - 2} - 1). \end{aligned}$$

Thus, from (2), we have for n a power of 2

$$\begin{aligned} T(n) &= n^{\lg 7} + cn^2(3/7)(n^{\lg 7 - 2} - 1) \\ &= (1 + 3c/7)n^{\lg 7} - (3c/7)n^2 \\ &= \Theta(n^{\lg 7}), \end{aligned}$$

where we note $\lg 7 = 2.81... > 2$.

When n is not a power of 2, observe that $T(n)$ is a nondecreasing function of n. That is $T(n') \geq T(n)$ if $n' \geq n \geq 1$. To show that, it suffices to prove $T(n+1) \geq T(n)$ inductively. Clearly $T(2) \geq T(1) = 1$. Inductively, let $n > 1$ and assume that we have established $T(n'+1) \geq T(n')$ for all $n' < n$. Then by (1) we have

$$\begin{aligned} T(n) &= 7T(\lceil n/2 \rceil) + cn^2, \\ T(n+1) &= 7T(\lceil (n+1)/2 \rceil) + c(n+1)^2. \end{aligned} \tag{3}$$

Note that $\lceil (n+1)/2 \rceil$ is either $\lceil n/2 \rceil$ or $\lceil n/2 \rceil + 1$. Furthermore, $\lceil (n+1)/2 \rceil < n+1$. Thus, we have by induction hypothesis $T(\lceil (n+1)/2 \rceil) \geq T(\lceil n/2 \rceil)$. This, together with (3), implies $T(n+1) \geq T(n)$, which completes the inductive proof that $T(n)$ is nondecreasing as a function of n.

Therefore, when n is not a power of 2, we have $T(2^{\lfloor \lg n \rfloor}) \leq T(n) \leq T(2^{\lceil \lg n \rceil})$. Note that $T(2^{\lceil \lg n \rceil}) = O((2^{\lceil \lg n \rceil})^{\lg 7}) = O((2^{1+\lg n})^{\lg 7}) = O((2 \cdot 2^{\lg n})^{\lg 7}) = O(n^{\lg 7})$, and similarly, $T(2^{\lceil \lg n \rceil}) = \Omega(n^{\lg 7})$. This implies that there exist constants $c, c' > 0$ such that $cn \leq T(n) \leq c'n$ for all $n \geq 1$. This proves that $T(n) = \Theta(n^{\lg 7})$.

A second approach is to guess what the right answer is, and then proceed to verify it by induction. We start with a simpler problem. Suppose the recurrence relation is $f(1) = 1$, and $f(n) = f(n-1) + 1 + 1/n$. Make the guess $f(n) = \Theta(n)$. To verify the guess, we can try to prove for example $f(n) \leq 3n$ and $f(n) \geq n$. Take the upper bound first. Clearly, $f(1) = 1 \leq 3 \cdot 1$, so the base case $n = 1$ is true. Inductively, let $n > 1$ and assume that we have proved $f(n') \leq 3n'$ for all $n' < n$. Then using the recurrence and the induction hypothesis for $n' = n - 1$ we have

$$\begin{aligned} f(n) &= f(n-1) + 1 + 1/n \\ &\leq 3(n-1) + 1 + 1/n \\ &\leq 3n. \end{aligned}$$

This proves $f(n) \leq 3n$ for all $n \geq 1$. Similarly, one can prove inductively $f(n) \geq n$ for all $n \geq 1$. That's what we mean by *guess and verify*.

The above approach is most useful when the recurrence relation is sufficiently complex that it is hard to solve exactly even for special values. For example, if we have $T(i) = 1$ for $1 \leq i \leq 9$, and $T(n) = 7T(\lceil n/2 \rceil + \lfloor \lg n \rfloor) + cn^2$ for $n > 9$. It is hard to find a special sequence for which we can solve the

problem exactly. An educated guess would be $T(n) = \Theta(n^{\lg 7})$ since the extra $\lg n$ is asymptotically very small compared to the n/2 in the recurrence relation.

One important point about the guess-and-verify approach. Sometimes just knowing the right answer is not enough. For inductive proofs, the *form* of the conjectured answer is also important. Take the Strassen recurrence (1). Suppose you guess $T(n) \leq 100(1+c)n^{\lg 7}$ and proceed with your induction proof. The base case $n = 1$ is clearly true. Inductively, let $n > 1$ and assume that the upper bound has been proved for all values less than n. Consider for simplicity the case n being even. Then using the recurrence and the induction hypothesis, we have

$$\begin{aligned} T(n) &= 7T(\lceil n/2 \rceil) + cn^2 \\ &\leq 7 \cdot 100(1+c)(n/2)^{\lg 7} + cn^2 \\ &= 100(1+c)n^{\lg 7} + cn^2. \end{aligned}$$

This unfortunately is not what we want to prove.

To carry out the proof successfully, choosing the right form is important. We make the guess $T(n) \leq (1+c)(100n^{\lg 7} - 20n^2)$, and proceed to prove it inductively. The base case $n = 1$ is clearly true. Inductively, the proof now looks like (if n is even)

$$\begin{aligned} T(n) &= 7T(\lceil n/2 \rceil) + cn^2 \\ &\leq 7 \cdot (1+c)(100(n/2)^{\lg 7} - 20(n/2)^2) + cn^2 \\ &= (1+c)100n^{\lg 7} - (1+c)(140/4)n^2 + cn^2 \\ &\leq (1+c)100n^{\lg 7} - (1+c)34n^2 \\ &\leq (1+c)(100n^{\lg 7} - 20n^2). \end{aligned}$$

Some additional adjustments need to be made before the inductive proof can go through when n is odd. We will not get into that at this time. For the moment, we only want to make the point that sometimes you have to choose the right form to carry out an inductive proof.

Problem E (c) in Homework Set No. 1 is a case in point, for which even after you have guessed the correct asymptotic form of the answer, you need to choose carefully the right expression in order to carry out a successful inductive verification.