Finding Θ for Recurrences

- We used an "iterate and cancel" method to solve recurrences.
- The textbook does not discuss solving recurrences.
- However, it discusses three methods for finding big-Θ for recurrences, which is good enough for our work.
- The first is the substitution method.
  - It is basically "guess and check."
  - But easier because you only have to guess big-Θ.
- The second is the recursion tree method.
  - This is essentially a pictorial match to our iterative method.
  - It may help generating guesses, but definitely needs to be proved using induction.
- The third is the master theorem.

Substitution Method

- Prove that $T(n) = O(n \lg n)$ for the following recurrence:

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

- Basically we guess the form of the element of $O(n \lg n)$, up to some constants, and then try to prove it works by induction.
- There are a number of possible guesses:

$$\begin{aligned}
T(n) &\leq cn \lg n \\
T(n) &\leq cn \lg n + dn \\
T(n) &\leq cn \lg n + d \lg n \\
T(n) &\leq cn \lg n + d \lg n + bn \\
T(n) &\leq cn \lg n + d\sqrt{n}
\end{aligned}$$

- We have to hope we hit on the right guess fairly quickly.
- Working with one guess can inform the next guess.

Substitution Method

- Prove that $T(n) = O(n \lg n)$ for the following recurrence:

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

- Guess that $T(n) \leq cn \lg n$ for some c and prove it by induction.
- Assume that $T(x) \leq cx \lg x$ for all $x < n$ and prove:

$$\begin{aligned}
T(n) &= 2T(n/2) + n && \text{by recurrence} \\
&\leq 2(c(n/2)\lg(n/2)) + n && \text{by inductive hypothesis} \\
&= cn \lg(n/2) + n \\
&= cn \lg n - cn \lg 2 + n \\
&= cn \lg n - cn + n \\
&\leq cn \lg n && \text{for } c \geq 1
\end{aligned}$$

- Can also prove $T(n) = \Omega(n \lg n)$ the same way.
- So, $T(n) = \Theta(n \lg n)$.

Base case

- Note that we didn't bother with the base case for induction.
- We can always choose a constant such that it's bigger than $T(1)$, which is also a constant.

Substitution method, guessing wrong element of $O(n)$

- Prove that $T(n) = O(n)$ for

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

- Guess that $T(n) \leq cn$.
- Assume that $T(x) \leq cx$ for $x < n$.

$$\begin{aligned} T(n) &= 2T(n/2) + 1 \\ &\leq 2(cn/2) + 1 \\ &= cn + 1 \end{aligned}$$

- This is *not* good enough, even though $cn + 1 = O(n)$.
- The problem is this is *one* step in an inductive proof, and if we add 1 for every step from 1 to n, we get more than $O(1)$.

Try a more general guess

- Prove that $T(n) = O(n)$ for

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

- Guess that $T(n) \leq cn + d$.
- Assume $T(x) \leq cx + d$ for $x < n$.

$$\begin{aligned}
T(n) &= 2T(n/2) + 1 \\
&\leq 2(cn/2 + d) + 1 \\
&= cn + 2d + 1 \\
&\leq cn \qquad\qquad d < -1
\end{aligned}$$

- This works, because we are free to pick any value for c and d!
- Note that we must pick *one* value for c and d for all inductive steps.

Changing Variables

- Some very difficult recurrences can be solved simply with change of variables.
- Example:

$$T(n) = 2T(\sqrt{n}) + \lg n$$

- Use the following substitutions:

$$\begin{aligned} m &= \lg n \\ T(2^m) &= 2T(2^{m/2}) + m \\ S(m) &= T(2^m) \\ S(m) &= 2S(m/2) + m \end{aligned}$$

- This can be solved to show that $S(m) = O(m \lg m)$, therefore

$$T(n) = T(2^m) = S(m) = O(m \lg m) = O(\lg n \lg \lg n)$$

The Jeopardy Method

- In order to practice solving recurrences, we need a lot of examples.
- It's difficult to know in advance whether a recurrence has a simple solution.
- We can solve this (tactical) problem by inventing our own recurrences, by starting with the *solution* to the recurrence (a function definition) and then discovering what recurrence it is a solution to.
- We can call this the "Jeopardy Method" of finding recurrences.

The Jeopardy Method

- Let's start with this function, for example:

$$f(n) = 3n^2 + 5$$

- We know the base case

$$f(1) = 8$$

- To get a recursion, do some simple math:

$$\begin{aligned} f(n-1) &= 3(n-1)^2 + 5 \\ &= 3n^2 - 6n + 3 + 5 \\ &= (3n^2 + 5) - 6n + 3 \\ &= f(n) - 6n + 3 \end{aligned}$$

- Therefore, our original function is the solution to the recurrence:

$$\begin{aligned} f(1) &= 8 \\ f(n) &= f(n-1) + 6n - 3 \end{aligned}$$

Check your question to the solution

- We can check this by substituting the function in both sides of the recurrence:

$$f(n) = 3n^2 + 5$$

$$f(n) = f(n-1) + 6n - 3$$

$$\begin{aligned} 3n^2 + 5 &= 3(n-1)^2 + 5 + 6n - 3 \\ &= 3n^2 - 6n + 3 + 5 + 6n - 3 \\ &= 3n^2 + 5 \end{aligned}$$

Equations and recurrences

- Let's try a different recurrence for the same function.

$$f(n) = 3n^2 + 5$$

- Let's try, $f(n/2)$:

$$\begin{aligned} f(n/2) &= 3(n/2)^2 + 5 \\ &= (3/4)n^2 + 5 \\ &= 3n^2 + 5 - (9/4)n^2 \\ &= f(n) - (9/4)n^2 \end{aligned}$$

- Our original function is also the solution to the recurrence:

$$\begin{aligned} f(1) &= 8 \\ f(n) &= f(n/2) + (9/4)n^2 \end{aligned}$$

Check your solution

- Now let's check this with our original equation plugged into both sides:

$$f(n) = 3n^2 + 5$$

$$f(n) = f(n/2) + (9/4)n^2$$

$$\begin{aligned} 3n^2 + 5 &= 3(n/2)^2 + 5 + (9/4)n^2 \\ &= (3/4)n^2 + 5 + (9/4)n^2 \\ &= 3n^2 + 5 \end{aligned}$$

- So, again, we have verified that our original function is a solution to this recurrence.

Substitution method, random example

- Show that $f(n) = O(n^2)$.

$$f(n) = f(n-1) + 6n - 3$$

- Assume, by inductive hypothesis, that $f(n-1) \leq c(n-1)^2$.
- Try to prove that $f(n) \leq cn^2$.

$$\begin{aligned} f(n) &= f(n-1) + 6n - 3 \\ &\leq c(n-1)^2 + 6n - 3 \\ &= cn^2 - 2cn + c + 6n - 3 \\ &= cn^2 + (6-2c)n + (c-3) \end{aligned}$$

- Now we just have to pick c and n_0 such that

$$(6-2c)n + (c-3) \leq 0$$

Picking c and n_0

- We want to assure

$$(6-2c)n+(c-3)\leq 0$$

- If we pick $c>3$ then

$$(6-2c)<0$$

- Therefore

$$(6-2c)n+(c-3)\leq 0 \iff n\geq \frac{(3-c)}{(6-2c)}$$

- So, let

$$n_0=\left\lceil \frac{(3-c)}{(6-2c)} \right\rceil$$

Substitution method, big-Ω

- To prove $\Theta(n^2)$ we also need $\Omega(n^2)$.

$$f(n) = f(n-1) + 6n - 3$$

- Suppose $f(n-1) \geq c(n-1)^2$ and prove $f(n) \geq cn^2$

$$\begin{aligned}
f(n) &= f(n-1) + 6n - 3 \\
&\geq c(n-1)^2 + 6n - 3 \\
&= cn^2 - 2cn + c + 6n - 3 \\
&= cn^2 + (6-2c)n + (c-3) \\
&\geq cn^2
\end{aligned}$$

- So long as $c < 3$, and sufficiently large n.
- Since $f(n) = O(n^2)$ and $f(n) = \Omega(n^2)$, $f(n) = \Theta(n^2)$.