Try backsubstituting until you know what is going on

Also known as the iteration method. Plug the recurrence back into itself until you see a pattern.

*Example:* \( T(n) = 3T([n/4]) + n. \)

Try backsubstituting:

\[
T(n) = n + 3\lfloor n/4 \rfloor + 3T(\lfloor n/16 \rfloor)
\]
\[
= n + 3\lfloor n/4 \rfloor + 9(\lfloor n/16 \rfloor + 3T(\lfloor n/64 \rfloor))
\]
\[
= n + 3\lfloor n/4 \rfloor + 9\lfloor n/16 \rfloor + 27T(\lfloor n/64 \rfloor)
\]

The \((3/4)^n\) term should now be obvious.

Although there are only \(\log_4 n\) terms before we get to \(T(1)\), it doesn’t hurt to sum them all since this is a fast growing geometric series:

\[
T(n) \leq n \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i + \Theta(n^{\log_4 3} \times T(1))
\]

\[
T(n) = 4n + o(n) = O(n)
\]
Recursion Trees

Drawing a picture of the backsubstitution process gives you a idea of what is going on.

We must keep track of two things – (1) the size of the remaining argument to the recurrence, and (2) the additive stuff to be accumulated during this call.

*Example:* \( T(n) = 2T(n/2) + n^2 \)

![Recursion Tree Diagram]

The remaining arguments are on the left, the additive terms on the right.

Although this tree has height \( \lg n \), the total sum at each level decreases geometrically, so:

\[
T(n) = \sum_{i=0}^{\infty} n^2/2^i = n^2 \sum_{i=0}^{\infty} 1/2^i = \Theta(n^2)
\]

The recursion tree framework made this much easier to see than with algebraic backsubstitution.