Recursion on Slightly Smaller Input

Recall recursive Insertion Sort:

$\textsc{InsertionSort}(A[1..n])$
  if $n > 1$
    $\textsc{InsertionSort}(A[1..n-1])$
    $\textsc{InsertLast}(A[1..n])$

$T(n)$, the worst-case running time for n, is the sum of

- testing if $n > 1$: $\Theta(1)$
- recursive call: $T(n-1)$
- calling $\textsc{InsertLast}$: $\Theta(n)$

We see that $T(n) \approx T(n-1) + n$ and thus

$$T(n) \approx \sum_{i=1}^{n} i \in \Theta(n^2)$$

just as the iterative version; no new techniques needed!

Recursion on Smaller Chunks of Input

Consider another sorting algorithm

$\textsc{MergeSort}(A[1..n])$
  if $n > 1$
    $m \leftarrow \lfloor n/2 \rfloor$
    $\textsc{MergeSort}(A[1..m])$
    $\textsc{MergeSort}(A[m+1..n])$
    $B[1..n] \leftarrow \textsc{Merge}(A[1..m], A[m+1..n])$
    $\textsc{copy}(B[1..n], A[1..n])$

$T(n)$, the running time for n, is the sum of

- the two recursive calls: $2 \cdot T(\frac{n}{2})$
- the call to $\textsc{Merge}$: $\Theta(n)$
- the call to $\textsc{Copy}$: $\Theta(n)$

We get the recurrence

$$T(n) = 2T(\frac{n}{2}) + \Theta(n)$$

How do we solve that recurrence?

General Recurrences

Assume the body of recursive function

- makes a recursive calls
- each of a size that is a factor b smaller than original call ($b > 1$)
- spends $f(n)$ time apart from recursive calls

Then we have the recurrence

$$T(n) = aT(\frac{n}{b}) + f(n)$$

where

- we may choose to also specify $T(n)$ for "small" n
- we may wrap floor $\lfloor\rfloor$ or ceiling $\lceil\rceil$ around $\frac{n}{b}$

but the asymptotic behavior of the solution will not depend on that.

Solving Recurrences

To solve a given recurrence, we may use

1. the Substitution Method
   - + can handle almost all recurrences
   - - requires some guesswork
2. the Master Theorem
   - + no guessing involved
   - - may not work for non-standard recurrences

Mathematical Induction

To prove that $P(n)$ holds for all natural numbers $n \geq 0$, it suffices to

1. prove $P(0)$
2. prove for arbitrary $m \geq 0$ that $P(m)$ implies $P(m+1)$.

Alternatively, only one obligation:

1. prove for arbitrary $m \geq 0$ that if $P(n)$ for all $n < m$ then $P(m)$

Substitution Method

Given recurrence

$$T(n) = \dots$$

we must

1. guess a g such that we expect that $T(n) \in O(g(n))$.
2. try to prove that (for unspecific c):

$$T(n) \le c \cdot g(n) \text{ for all } n \ge \dots$$

   and if successful, a suitable c will emerge

Similarly, one may prove $T(n) \in \Omega(g(n))$.

Substitution Method, Example 1

Consider the recurrence

$$T(n) = 2 \cdot T(\lfloor \frac{n}{2} \rfloor) + n^2$$

Guess:

$$T(n) \in O(n^2)$$

To verify this, we try to prove, for yet unknown c,

$$T(n) \leq cn^2$$

by induction in $n \geq 1$

Substitution Method, Example 2

Recall the MERGE SORT recurrence (with floor inserted)

$$T(n) = 2 \cdot T(\lfloor \frac{n}{2} \rfloor) + n$$

Guess:

$$T(n) \in O(n \lg(n))$$

To verify that, we try to prove, for yet unknown c,

$$T(n) \leq cn \lg(n)$$

by induction in $n \geq 2$.

- for inductive step we need $n \geq 4$ so that

$$2 \leq \lfloor \frac{n}{2} \rfloor < n$$

- the base cases are now $n = 2$, $n = 3$

Alternatively: prove $T(n) \leq cn \lg(2n)$ for $n \geq 1$.

Substitution Method, Example 3

Now consider

$$T(n) = 2 \cdot T(\lfloor \frac{n}{2} \rfloor) + 1$$

Guess:

$$T(n) \in O(n)$$

We now try to prove, for yet unknown c,

$$T(n) \leq cn$$

by induction in $n \geq 1$.

- the inductive step is for $n \geq 2$ but does not work out.

Instead we must prove something stronger:

$$T(n) \leq cn - d \text{ with } d \geq 0$$

Solve by Unfolding: Special Case

Recall that we used the substitution method to show that

$$T(n) = n + 2 \cdot T(\lfloor \frac{n}{2} \rfloor)$$

has solution $T(n) \in O(n \lg(n))$ and we can even show

$$T(n) \in \Theta(n \lg(n))$$

Another way to see that this is a solution is to unfold:

$$\begin{aligned}
T(n) &= n + 2T(\frac{n}{2}) \\
&= n + 2(\frac{n}{2} + 2T(\frac{n}{4})) \\
&= n + 2(\frac{n}{2} + 2(\frac{n}{4} + 2T(\frac{n}{8}))) \\
&\approx \overbrace{n + n + n + \ldots + n}^{\lg n \text{ times}} + n \cdot T(1) \\
&\approx n \lg(n)
\end{aligned}$$

Solve by Unfolding: General Case

The general recurrence $T(n) = n^q + aT(\frac{n}{b})$ unfolds to

$$T(n) \approx (1 + c + c^2 + \ldots + c^{\log_b(n)})n^q \text{ where } c = \frac{a}{b^q}$$

and we now split into 3 cases:

- If $c < 1$ then the first term dominates: $T(n) \approx \frac{1}{1-c} n^q$ and thus $T(n) \in \Theta(n^q)$
- If $c = 1$ then $T(n) \approx \log_b(n) n^q$ and thus $T(n) \in \Theta(n^q \lg(n))$
- if $c > 1$ then the last term dominates: $T(n) \approx c^{\log_b(n)} n^q$ and since

$$c^{\log_b(n)} n^q = \frac{a^{\log_b(n)}}{(b^{\log_b(n)})^q} n^q = a^{\log_b(n)} = n^{\log_b(a)}$$

  we get $T(n) \in \Theta(n^{\log_b(a)})$

The above reasoning can be made rigorous!

General Recipe for Solving Recurrences

For the general recurrence $T(n) = aT(\frac{n}{b}) + n^q$ we showed

$$T(n) \approx (1 + c + c^2 + \ldots + c^{\log_b(n)})n^q$$

with cases depending on $c = \frac{a}{b^q}$:

- If $c < 1$ then $T(n) \in \Theta(n^q)$
- If $c = 1$ then $T(n) \in \Theta(n^q \lg(n))$
- if $c > 1$ then $T(n) \in \Theta(n^{\log_b(a)})$

With $r = \log_b(a)$ and thus $b^r = a$, we can rephrase:

- If $q > r$ then $T(n) \in \Theta(n^q) = \Theta(n^{\max(q,r)})$
- If $q = r$ then $T(n) \in \Theta(n^q \lg(n))$
- If $q < r$ then $T(n) \in \Theta(n^r) = \Theta(n^{\max(q,r)})$

Master Theorem (One Version)

Recall: for $T(n) = aT(\frac{n}{b}) + n^q$, with $r = \log_b(a)$ we have

- If $q > r$ then $T(n) \in \Theta(n^q)$
- If $q = r$ then $T(n) \in \Theta(n^q \lg(n))$
- If $q < r$ then $T(n) \in \Theta(n^r)$

This can be generalized into the Master Theorem: if

$$T(n) = aT(\frac{n}{b}) + f(n) \text{ for } n \geq b$$

where

- b is an integer with $b \geq 2$ (b real > 1 also works)
- either $\lfloor \; \rfloor$ or $\lceil \; \rceil$ is wrapped around $\frac{n}{b}$
- $T(n)$ is eventually non-decreasing (prove separately)

then with $r = \log_b a$ we have for $X \in \{O, \Omega, \Theta\}$

1. if $f(n) \in X(n^q)$ with $q > r$ then $T(n) \in X(n^q)$
2. if $f(n) \in X(n^q)$ with $q = r$ then $T(n) \in X(n^r \lg n)$
3. if $f(n) \in O(n^q)$ with $q < r$ then $T(n) \in \Theta(n^r)$.

Master Theorem Versions

- *Cormen*'s Theorem 4.1 (p.102)
- *Howell*'s Theorem 3.32
- Wikipedia

They never produce conflicting answers, but are phrased differently, with some more applicable than others:

- Cormen and Wikipedia
  - use reverse case ordering
  - allow b to be real > 1
- Howell's version of case 2 is very general so as to be applicable for a wide range of recurrences
  - a few special cases are handled also by Cormen and Wikipedia

A Slightly Non-Standard Recurrence

Consider the recurrence ($a = 2, b = 2, r = 1$)

$$T(n) = 2T(\frac{n}{2}) + f(n)$$

and assume $f(n) \in \Theta(n \lg(n))$.

- for all real $q > 1$, since $f(n) \in O(n^q)$, we know $T(n) \in O(n^q)$.
- obviously, $T(n) \in \Omega(n \lg(n))$.

Thus for all real $q > 1$,

$T(n)$ is sandwiched between $n \lg(n)$ and n^q

The other Master Theorem versions have a stronger case 2:

- Cormen and Wikipedia immediately yield $T(n) \in \Theta(n \lg(n) \lg(n))$.
- Howell can be instantiated to also yield $T(n) \in \Theta(n \lg(n) \lg(n))$.

A Somewhat Non-Standard Recurrence

Consider the recurrence ($a = 2, b = 2, r = 1$)

$$T(n) = 2T(\frac{n}{2}) + f(n)$$

and assume $f(n) \in \Theta(\frac{n}{\lg(n)})$.

- since $f(n) \in O(n)$, we know $T(n) \in O(n\lg(n))$.
- obviously, $T(n) \in \Omega(\frac{n}{\lg(n)})$.

Thus $T(n)$ is sandwiched between $\frac{n}{\lg(n)}$ and $n\lg(n)$

and none of the listed versions of the Master Theorem can give a more precise bound except for an extended version of Wikipedia's which gives

$$T(n) \in \Theta(n\lg(\lg(n))).$$

A Very Non-Standard Recurrence

Now consider the recurrence

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

which does not appear to fit any Master Theorem. But let $S(k) = T(2^k)$. We then have

$$S(k) = T(2^k) = 2T(\sqrt{2^k}) + \lg(2^k) = 2T(2^{\frac{k}{2}}) + k$$

and hence the recurrence

$$S(k) = 2S(\frac{k}{2}) + k$$

and thus

$$S(k) \in \Theta(k \lg(k))$$

which we can translate into a solution for T:

$$T(n) = S(\lg(n)) \in \Theta(\lg(n) \lg(\lg(n))).$$

You may (should) feel a bit uneasy, but it is possible to verify it by the substitution method.