

Announcements



- **Reading Assignments: Chapter 4 & 5
(Textbook: CLRS)**
- **Reminder: Homework #1 is due this
Thursday, September 8, 2005**
- **TA: Suddha Basu (SN008)
Office Hours: Mon/Wed 2:00-3:30pm**

Divide-and-Conquer



- Recursive in structure
 - *Divide* the problem into several smaller sub-problems that are similar to the original but smaller in size
 - *Conquer* the sub-problems by solving them recursively. If they are small enough, just solve them in a straightforward manner.
 - *Combine* the solutions to create a solution to the original problem

An Example: Merge Sort



- *Divide*: Divide the n -element sequence to be sorted into two subsequences of $n/2$ elements each
- *Conquer*: Sort the two subsequences recursively using merge sort.
- *Combine*: Merge the two sorted subsequences to produce the sorted answer.

Merge-Sort (A, p, r)



INPUT: a sequence of n numbers stored in array A

OUTPUT: an ordered sequence of n numbers

1. if $p < r$
2. then $q \leftarrow \lfloor (p+r)/2 \rfloor$
3. Merge-Sort (A, p, q)
4. Merge-Sort ($A, q+1, r$)
5. Merge (A, p, q, r)

Analysis of Merge Sort



- Divide: computing the middle takes $\Theta(1)$
- Conquer: solving 2 sub-problem takes $2T(n/2)$
- Combine: merging n -element takes $\Theta(n)$
- Total:

$$T(n) = \Theta(1) \quad \text{if } n = 1$$

$$T(n) = 2T(n/2) + \Theta(n) \quad \text{if } n > 1$$

$$\Rightarrow T(n) = \Theta(n \lg n) \text{ (CLRS/Chapter 4)}$$

Recurrence Relations



- Recurrences (Chapter 4)
 - Substitution Method
 - Iteration Method
 - Master Method
- Arising from Divide and Conquer (e.g. MERGE-SORT)

$$T(n) = \Theta(1) \quad \text{if } n \leq c$$

$$T(n) = a T(n/b) + D(n) + C(n) \quad \text{otherwise}$$

Substitution Method



- Guessing the form of the solutions, then using mathematical induction to find the constants and show the solution works.
- It works well when it is easy to guess. But, there is no general way to guess the correct solution.

An Example



- **Solve:**
$$\begin{aligned} T(n) &= 3T(\lfloor n/3 \rfloor) + n \\ T(n) &\leq 3c \lfloor n/3 \rfloor \lg \lfloor n/3 \rfloor + n \\ &\leq c n \lg (n/3) + n \\ &= c n \lg n - c n \lg 3 + n \\ &= c n \lg n - n (c \lg 3 - 1) \\ &\leq c n \lg n \end{aligned}$$

* The last step is true for $c \geq 1 / \lg 3$.

Making a Good Guess



- Guessing a similar solution to the one that you have seen before
 - $T(n) = 3T(\lfloor n/3 \rfloor + 5) + n$ similar to $T(n) = 3T(\lfloor n/3 \rfloor) + n$
when n is large, the difference between $n/3$ and $(n/3 + 5)$ is insignificant
- Another way is to prove loose upper and lower bounds on recurrence and then reduce the range of uncertainty.
 - Start with $T(n) = \Omega(n)$ & $T(n) = O(n^2) \Rightarrow T(n) = \Theta(n \log n)$

Subtleties



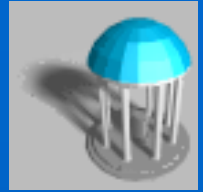
- When the math doesn't quite work out in the induction, try to adjust your guess with a lower-order term. For example:
 - We guess $T(n) \leq O(n)$ for $T(n) = 3T(\lfloor n/3 \rfloor) + 4$, but we have $T(n) \leq 3c \lfloor n/3 \rfloor + 4 = c n + 4$
 - New guess is $T(n) \leq c n - b$, where $b \geq 0$
 $T(n) \leq 3(c \lfloor n/3 \rfloor - b) + 4 = c n - 3b + 4 = c n - b - (2b - 4)$
Therefore, $T(n) \leq c n - b$, if $2b - 4 \geq 0$ or if $b \geq 2$

Changing Variables



- Use algebraic manipulation to turn an unknown recurrence similar to what you have seen before.
 - Consider $T(n) = 2T(\lfloor n^{1/2} \rfloor) + \lg n$
 - Rename $m = \lg n$ and we have
$$T(2^m) = 2T(2^{m/2}) + m$$
 - Set $S(m) = T(2^m)$ and we have
$$S(m) = 2S(m/2) + m \Rightarrow S(m) = O(m \lg m)$$
 - Changing back from $S(m)$ to $T(n)$, we have
$$T(n) = T(2^m) = S(m) = O(m \lg m) = O(\lg n \lg \lg n)$$

Avoiding Pitfalls



- Be careful not to misuse asymptotic notation.
For example:

- We can falsely prove $T(n) = O(n)$ by guessing $T(n) \leq c n$ for $T(n) = 2T(\lfloor n/2 \rfloor) + n$

$$T(n) \leq 2c \lfloor n/2 \rfloor + n$$

$$\leq c n + n$$

$$= O(n) \Leftarrow \text{Wrong!}$$

- The err is that we haven't proved $T(n) \leq c n$

Exercises



- Solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$
- Solution of $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ is $O(n \lg n)$
- Solve $T(n) = 2T(n^{1/2}) + 1$ by making a change of variables. Don't worry whether values are integral.