Slide 1

Administrivia

- Homework deadlines extended — Homework 4 due Monday, Homework 5 next Friday.
- Reminder: Quiz 3 Friday.
- Midterm changed to 3/24 (Friday after spring break). Review sheet will be on Web.

Slide 2

What To Write Down for Proofs

- First write down what you are given, what you are to prove — possibly turning words into formulas.

  Example — "the sum of consecutive squares . . ." becomes $(n^2 + (n+1)^2$.
- Then write down a logical argument, starting with what you are given and ending with what you are to prove.

  Similar to what we did with formal logic, but less structured.
- Examples — answers in back of textbook, examples from board in class.

Slide 3

Recursive Definitions — Review

- Recall examples from last time — recursive definitions of sequences, sets, operations, algorithms.
- For sets, notice that this means that to claim that something is in the set you need to be able to show that it's either a base case or can be obtained from a base case by applying one of the "rules" that define the set.

Slide 4

Recursive Algorithms, More Examples

- Two good examples in text — selection sort and binary search.
- Another example — "quicksort".

```
// pre:  i, j are valid indices for L
// post:  L(i) through L(j) are "sorted"
qsort(list L, index i, index j)
        if (i >= j)
                return
        else
                elem pivot = L(i)
                // rearrange L(i+1) through L(j) s.t.:
                //      L(i) .. L(m-1) <= pivot
                //      L(m) = pivot
                //      L(m+1) .. L(j) >= pivot
                index m = split(pivot, L, i, j)
                qsort(L, i, m-1)
                qsort(L, m+1, j)
end qsort
```

Slide 5

Recurrence Relations

- Recall the silly example of defining a sequence recursively:

$$\begin{aligned} S(1) &= 1 \\ S(n) &= S(n-1) \times 10, \quad \text{for } n > 1 \end{aligned}$$

  Expanding out some terms, it seems fairly obvious that an equivalent definition would be $S(n) = 10^{n-1}$ — a "closed-form solution" to the recurrence relation given in the second line of the definition.

- We'll look at various ways to get from a recursive definition to a closed-form one, because the latter are easier to compute, but sometimes it will be much easier to write down the definition recursively.

Slide 6

Solving Recurrence Relations, Continued

- For the silly example

$$\begin{aligned} S(1) &= 1 \\ S(n) &= S(n-1) \times 10, \quad \text{for } n > 1 \end{aligned}$$

  we guessed a solution of $S(n) = 10^{n-1}$. Can we verify that this is the same as the recursive definition? yes, via a proof by induction ...

- Call this method "expand, guess, verify".

- Try another example — section 2.4 problem 75.

Minute Essay

- We started a proof by induction that the guessed closed-form solution for problem 75 ($T(n) = 2^n - 1$) matches the recursive definition:
  Base case: $2^1 - 1 = 1 = T(1)$
  Inductive step: Assume $T(k) = 2^k - 1$ and show $T(k+1) = 2^{k+1} - 1$.
  Complete the proof.

Slide 7

Minute Essay Answer

- $T(k+1) = 2T(k) + 1$ (using recursive definition).
  $2T(k) + 1 = 2(2^k - 1) + 1$ (using inductive hypothesis).
  $2(2^k - 1) + 1 = 2^{k+1} - 2 + 1 = 2^{k+1} + 1$, which completes the proof.

Slide 8