

**4.3-9**

Solve the recurrence  $T(n) = 3T(\sqrt{n}) + \log n$  by making a change of variables. Your solution should be asymptotically tight. Do not worry about whether values are integral.

---

**4.4 The recursion-tree method for solving recurrences**

Although you can use the substitution method to provide a succinct proof that a solution to a recurrence is correct, you might have trouble coming up with a good guess. Drawing out a recursion tree, as we did in our analysis of the merge sort recurrence in Section 2.3.2, serves as a straightforward way to devise a good guess. In a **recursion tree**, each node represents the cost of a single subproblem somewhere in the set of recursive function invocations. We sum the costs within each level of the tree to obtain a set of per-level costs, and then we sum all the per-level costs to determine the total cost of all levels of the recursion.

A recursion tree is best used to generate a good guess, which you can then verify by the substitution method. When using a recursion tree to generate a good guess, you can often tolerate a small amount of “sloppiness,” since you will be verifying your guess later on. If you are very careful when drawing out a recursion tree and summing the costs, however, you can use a recursion tree as a direct proof of a solution to a recurrence. In this section, we will use recursion trees to generate good guesses, and in Section 4.6, we will use recursion trees directly to prove the theorem that forms the basis of the master method.

For example, let us see how a recursion tree would provide a good guess for the recurrence  $T(n) = 3T(\lfloor n/4 \rfloor) + \Theta(n^2)$ . We start by focusing on finding an upper bound for the solution. Because we know that floors and ceilings usually do not matter when solving recurrences (here’s an example of sloppiness that we can tolerate), we create a recursion tree for the recurrence  $T(n) = 3T(n/4) + cn^2$ , having written out the implied constant coefficient  $c > 0$ .

Figure 4.5 shows how we derive the recursion tree for  $T(n) = 3T(n/4) + cn^2$ . For convenience, we assume that  $n$  is an exact power of 4 (another example of tolerable sloppiness) so that all subproblem sizes are integers. Part (a) of the figure shows  $T(n)$ , which we expand in part (b) into an equivalent tree representing the recurrence. The  $cn^2$  term at the root represents the cost at the top level of recursion, and the three subtrees of the root represent the costs incurred by the subproblems of size  $n/4$ . Part (c) shows this process carried one step further by expanding each node with cost  $T(n/4)$  from part (b). The cost for each of the three children of the root is  $c(n/4)^2$ . We continue expanding each node in the tree by breaking it into its constituent parts as determined by the recurrence.