Please note that if a problem asks you to give an algorithm that runs in time $O(n)$, say that does some particular task, you have to show two things:

- You have to prove that your algorithm really does what it is supposed to do.
- You have to prove that the running time of the algorithm is $O(n)$.

1. The proof of Lemma 1.3 in the Lecture 12 handout is a proof by induction, but is presented a bit informally. What is the inductive hypothesis?

2. Exercise 22.2-2 (page 601).

3. Exercise 22.2-6 (page 602).

4. Exercise B.5-3 (page 1180). (The degree of a node in a rooted tree is by definition the number of its children.) In doing this problem, be sure to carefully state the inductive hypothesis. This will help you in constructing the proof.

5. Exercise 22.3-2 (page 610).

6. Exercise 22.3-8 (page 611). The notation $u.d$ in that problem refers to the “discovery time” or (as it is called in my notes) the “start time” for vertex $u$ in the depth-first walk.

7. Exercise 22.3-9 (page 612). (And here, $u.f$ is the “finish time” for node $u$.)

8. Exercise 22.4-1 (page 614).

9. Exercise 22.4-2 (page 614).