Review Questions for Reading #2

1. What is a specification of requirements for a computational problem?

   (a) What is a precondition?

   (b) What is a postcondition?

2. What is an algorithm?

3. What does it mean for a given algorithm (that supposedly solves a given computational problem, with a given specification of requirements) to be correct?
4. What *proof technique* can often be used to prove that a simple recursive algorithm is correct?

5. What is a *bound function for a recursive algorithm*?

6. Why are bound functions for recursive algorithms useful? In what way (if at all) are they related to the proofs of correctness of these algorithms?

7. What do you need to do in order to *prove* that a given function is a bound function for a given recursive algorithm?
8. What are assertions — and why are useful? In what way (if at all) are they related to a proof of correctness of a recursive algorithm?

9. Describe a part of Java or C++ (or both) that “supports” assertions or makes them useful, in some way.

10. What is a trace of execution of an algorithm, on a given input? How can this be useful?

11. What is a recursion tree for a recursive algorithm and a given input?