

Homework, Week 2, part A

October 24, 2013

1 Software process

The software development process has four steps. This week's homework will walk you through all four steps. The steps are: (1) analysis, (2) design, (3) implementation, and (4) testing. The process repeats all four steps, one after another, many times before the software is complete.

The purpose of analysis is to develop the **functional requirements** of the software – what the software is supposed to do. Suppose you were to develop a number guessing game, with numbers between 1 and 100, with the computer to guess the number you pick. Write down everything you can think of that the software should do. This might include things like (a) tell the computer that its guess was too high or low, and (b) check the guesses.

The purpose of design is to think about how the software will actually be written. Taking your functional requirements from the previous answer, develop a design for your addition test. This might include such things as (a) start the game, (b) stop the game, and (c) tell when an answer is correct.

Make sure that you think hard about your answers and that you write them down. You will use these answers for the rest of the week when we implement the addition test. Barski has done some of this on page 22 of the book, but he did not distinguish between requirements and design. For example, steps 3 and 4 would be part of the functional requirements, while steps 1 and 2 would be part of the design.

2 Bisection algorithm

This week introduces a very important algorithm called *bisection search*. It's used when the task is *searching*, when the object to be found is *known*, and when the objects to be searched are *ordered*. You first look at the *middle* object, if that matches, you're done. If it does not match, you look at the middle of the *top half* or *bottom half*, depending on whether the object you want is greater or lesser than the object you just found. You continue halving the objects, depending on the comparison of the object you found with the object you want, until you are finished.

This week, you will write two variations of bisection search as a guess-the-number game. In one version, the computer will employ the search algorithm, and in the second version, the human will employ the search algorithm.