Study Guide 3

Big Ideas

- **Induction** – be familiar with establishing *base case* and *inductive step*.
  I. Base Case: $P(0)$ is true
  II. Inductive Step: Assume $P(n)$ for a specific n and show $P(n) \to P(n+1)$.
- **Strong Induction** – It's like normal induction, except we potentially assume multiple base cases and need to assume $P(0)$ through $P(n)$ are true to show that $P(n+1)$ is true.
- **Well-Ordering Principle** – If S is a set of one or more integers all greater than some fixed integer, then S has a least element.
- **Induction Application**: **Algorithms** – You can show an algorithm is "correct" by apply induction to variables that go through iterations of a loop. This involves use of a *loop invariant*, a predicate that is true before the loop and remains true after passing through the loop, and the *guard*, which is the name given to the predicate condition that keeps statements in the loop. We denote $I(n)$ to be the nth iteration of the loop invariant through the loop.
  I. Basis Step – Identical to base case. Check that the pre-condition is true before the loop, i.e. pre-condition implies $I(0)$ is true.
  II. Induction Step – Show that if loop invariant $I(n)$ and guard G are true before the loop, then $I(n+1)$ is true after the loop.
  III. Eventual Falsity of the Guard – Show that eventually G will become false (we don't want the loop to run infinitely!).
  IV. Correctness of Post-Condition – Check that the post-condition is true after the loop, i.e. if N is the smallest value for which $I(N)$ is true and the guard G becomes false, then the post-condition is also true.

  I'm aware this looks like a lot, but realistically it is identical to induction with the added steps of checking the loop doesn't run infinitely and that the end result isn't contradictory.

- **Recurrence Relations** – A recurrence relation is defined as a formula for a sequence that defines elements in terms of previous elements. For example:

  $$a_n = a_{n-1} + 3a_{n-2} - 7a_{n-3}$$

  relates values in the sequence to the previous three values.

- **Solving Second-Order Recurrence Relations** – For a recurrence relation of the form $a_n = Aa_{n-1} + Ba_{n-2}$, you can find sequences that satisfy the relation by solving the characteristic equation

  $$t^2 - At - B = 0$$

  The solution sequence would then be $a_n = t^n$ for $n \geq 0$.

For second-order recurrence relations with initial conditions a_0 and a_1, you can find an explicit formula by solving for constants C and D as follows:
- If r and s are *unique* roots of the characteristic equation:
$$a_n = Cr^n + Ds^n$$
- If r is a *repeated* root of the characteristic equation:
$$a_n = Cr^n + Dn \cdot r^n$$

- **Structural Induction** – Sets are defined recursively using a BASE group of elements, a list of RECURSION rules to create new elements, and a RESTRICTION that these are the only ways to form elements of the set. You can perform induction on the set to check if a condition is true by applying the recursion rules of the set to the base case of the induction.
- **Element Argument of Sets** – To show that $X \subseteq Y$, show that if you assume $x \in X$ arbitrarily, that $x \in Y$ as well.
- **Set Equality** – To show that sets $X = Y$, you must show $X \subseteq Y$ and $Y \subseteq X$.
- **Element Method for the Empty Set** – To show a set $X = \emptyset$, use proof by contradiction – assume $\exists x \in X$ and demonstrate this is a contradiction.
- **Power Sets** – The set of all subsets of a set X is called the *power set* of X and is denoted $\mathcal{P}(X)$.

Practice Textbook Problems

Section 5.2: #11, 16, 33-35

Section 5.3: #8, 11, 13, 19, 23, 26, 29

Section 5.4: #8-11, 17, 22, 23, 26, 32

Section 5.5: #1-5

Section 5.6: #3-5, 9-12, 17

Section 5.7: #1, 3-6

Section 5.8: #11-16, 20, 24

Section 5.9: #4-11

Section 6.1: #3-5, 12, 13, 22, 24, 27, 30, 31

Section 6.2: #7-9, 11-13, 16, 25-29, 38, 40