## Inductive Proof Guidelines

1. Define P(n). Note that P(n) is a **proposition**. Not a function. Therefore, if you say:

   P(n) = $3 + 3*5 + 3*5^2 + \ldots + 3*5^n = \frac{3(5^{n+1}-1)}{4}$, $n \geq 0$

   You have just defined a function, not a proposition.
   This is how to define a proposition correctly:

   P(n): $3 + 3*5 + 3*5^2 + \ldots + 3*5^n = \frac{3(5^{n+1}-1)}{4}$, $n \geq 0$

   Small difference, but means completely different things. After you define the proposition this way, do **not** say it is equal to anything besides true or false. Propositions can't equal anything else, they're boolean values.
   *Structural induction: usually the predicate will be of the following form: P(x): x has some property Q.*
2. Define n. What is its domain, bounds, etc.?
   *Structural Induction: typically you define x as a member of some recursively defined set S.*
3. Proof type, if not explicitly specified (drawing a distinction between induction or strong induction is not necessary, you can simply state "proof by induction" for either case, although specifying clearly is a good idea).
4. State base case(s) and show it/them to be true.
   *Structural induction: Show that P(x) is true for each x in the set specified by the basis step of the recursive definition of S.*
5. State the inductive hypothesis. Be sure to define the bounds and domain of any variables you use. **NOTE:** Do **not** re-use variables! It *may* be possible to do so correctly, but it is generally a bad idea.

   Example for regular induction: Assume P(k), $k \in \mathbb{Z}$, $k \geq 0$.

   Example for strong induction: Assume P(j), where $j, k \in \mathbb{Z}$, $k \geq 0$, and $0 \leq j \leq k$

   Since you have already defined P(n), you need not write out P(k) explicitly. However, doing so is often helpful.
   *Example for structural induction: Assume P(x) for all x currently in S at some arbitrary point in the recursive generation.*
6. State what you will prove (usually P(k + 1)).
   Example: I will show P(k + 1).
   You need not define P(k + 1), but you may find that helpful (I generally do for all but the simplest induction proofs).
   *Structural induction: Want to show that P(x') holds where x' is an arbitrary element to be generated using the recursive definition of S (where S is in the same state as it is in guideline 5).*
7. Body of inductive step: Prove P(k+1). If this means proving an equality, then start with one side of the equation and show it to be equal to the other. Remember that your proof should make use of the fact that P(k) is true; if you are proving an equality this might be accomplished by breaking an expression into a term that appeared in the statement of

P(k).
*Structural Induction: Typically you will show that if the elements used to generate x' have property Q (which is where the inductive hypothesis comes in), then the generation of x' ensures that x' has property Q.*

8. Explicitly state where you use the inductive hypothesis ("by IH" is sufficient).
9. Finish deriving the other side of P(k + 1).
10. Conclude. As usual, a simple QED is sufficient, but I highly recommend being a bit more verbose, as induction proofs have a more complicated structure than other proof types. This is how I usually wrap my induction proofs up: "I have shown that P(k) → P(k + 1), thus by induction P(n) is true for all n within the stated bounds. QED." This is for regular induction; the conditional for strong induction is a bit unwieldy, so I would probably abbreviate the above to just "Therefore, by strong induction, P(n) is true for all n within the stated bounds."

The usual rules for general proofs apply - you must cite *all* theorems (e.g. theorem 5 from section 3.4), and you must explicitly state givens and any other assumptions (e.g. "Given that x is odd" or "Assume that x is an odd number"). You need not cite definitions unless you feel that the proof would benefit from the citation. Remember, proofs are meant to be read and so should be clear and concise, without relying too much on the memory of the reader for obscure definitions and theorems.